Fair Wrapping for Black-box Predictions

Anonymous Author(s)
Affiliation
Address
email

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

Machine Learning has seen a dramatic increase of its impact over the past decade – enough that it has become a priority to control not just the accuracy, but also the bias of models' outputs [1, 14, 36]. If we take into account the numerous fairness targets and models that have been defined and / or refined [20] – sometimes excluding each other or in tension with accuracy, and the factor of energy and CO2 footprint of the domain [19, 29] – then the combinatorics of training accurate *and* fair models look non trivial. A suitable trend in the field tries to "decouple" both constraints as it seeks to post-process the outputs of *pretrained* (accurate) models to achieve a more fair output [36]. Post-processing may be the only option if we have no access to the model's training data/algorithm/hyperparameters (etc.).

In this cluster, 3 trends emerge: learning a new fair model close to the black-box, tweaking the output subject to fairness constraints, and exploiting sets of classifiers (§ 2). If the task is class probability estimation [25], the estimated black-box is an accurate but potentially unfair posterior $\eta_{\rm u} : \mathcal{X} \rightarrow [0,1]$ which neither can be opened nor trained further. The goal is then to learn a fair posterior $\eta_{\rm f}$ from it. A number of relevant *desiderata* can be considered for post-processing, including, **(flexibility)** of training to substantially different fairness metrics, **(proximity)** of learnt $\eta_{\rm f}$ to original $\eta_{\rm u}$, meaningful **(composability)** if *e.g.* $\eta_{\rm f}$ was later treated as a new black-box to be post-processed, **(explainability)** in auditing the post-processing, meaningful bounds of the post-processed classifier's **(complexity)** for generalization, algorithmic guarantees for **(convergence)**.

Our contribution explores a new solution to the post-processing problem, which learns a *correction* for fairness $\alpha : \mathcal{X} \rightarrow \mathbb{R}$ and defines $\eta_{\rm f}$ as a function of $\eta_{\rm u}$ and α. For this objective, we use a recently introduced theory which allows to fully correct a twist in posterior prediction through the minimization of specific *im*proper losses – unfairness being the twist. The approach also addresses the desiderata from three standpoints: analytical, representation, and algorithmic. From an analytical standpoint, we show that the correction yields convenient divergence bounds between $\eta_{\rm f}$ and $\eta_{\rm u}$, a convenient form for the Rademacher complexity of the class of $\eta_{\rm f}$, 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 exemplify the algorithm on three fairness metrics:

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

conditional value at risk, equality of opportunity, and statistical parity. Intuitively, our approach can be summarized as a method to utilize boosting to learn a α-tree to parameterize corrections to a blackbox such that it will adhere to fairness constraints. Experiments are provided against five baselines on readily available datasets. All proofs and many more experiments are in an Appendix denoted SI.

2 Related work

Post-processing models to achieve fairness is one out of three different categories of approaches to tackle the ML + fairness challenge [36, Section 6.2]. We focus on group fairness, which concerns itself with ensuring that the statistics of sub-population are similar. Although other notions of fairness exist, *e.g.* individual fairness [12], we limit our analysis to group fairness. We can segment this cluster further in three subsets: (I) approaches learning a new model with two constraints: being close to the pretrained model and being fair [16, 23, 32, 35]; (II) approaches biasing the output of the pretrained model at classification time, modifying observations to receive a more fair outcome [1, 14, 18, 21, 34, 35]; and a last one (III) consisting of exploiting sets of models to achieve fairness [11]. None of those approaches formulates substantial guarantees on all of the desiderata in the introduction. Some bring contributions applicable to more than two fairness notions [7, 32, 11, 35] **(flexibility)**, two of which provide the convenience of analytic conditions on new fairness notions to fit in the approach [32, 11], but for all of them the algorithmic price-tag is unclear [7, 11] or heavily depends on convex optimization routines [32]. [1, 35] provide strong guarantees regarding **(proximity)**, in terms of *consistency and generalization*. To our knowledge, our approach that corrects prediction unfairness through improper losses [30] is new.

3 Losses for class probability estimation

We first refresh concepts from the literature regarding class probability estimation and the α-loss.

Binary experiments and supervised learning Let $\mathcal{X}$ be a domain of observations, $\mathcal{Y} \doteq \{-1, 1\}$ labels and S is a sensitive attribute in $\mathcal{X}$. We assume that the modalities of S induce a partition of $\mathcal{X}$. $(\mathcal{X}, \mathrm{P})$ and $(\mathcal{X}, \mathrm{N})$ are measure spaces for "positive" and "negative" observations respectively, leaving implicit the σ-algebra, assumed to be the same everywhere. $(\mathcal{X} \times \{-1, 1\}, \mathrm{D})$ is is the joint distribution following the *binary task* $(\pi, \mathrm{P}, \mathrm{N})$ [25, Section 4], $\pi \doteq \mathbb{P}[\mathsf{Y} = 1]$ being the prior. $(\mathcal{X}, \mathrm{M})$ is a *mixture* measure space defined by $\mathrm{M} \doteq \pi \cdot \mathrm{P} + (1 - \pi) \cdot \mathrm{N}$. This notation can simply be intuitively though of as a decomposition of the joint space of features and labels. As is often assumed in ML, sampling is i.i.d.; we make no notational distinction between empirical and true measure to simplify exposure as most of our results would apply for both. Distinction shall be made when discussing generalization. Finally, $\eta \in [0, 1]^{\mathcal{X}}$ denotes a posterior that computes (an estimate of) $\mathbb{P}[\mathsf{Y} = 1|\mathsf{X}]$. In this paper, blue-boxed text is used to single out algorithmic nuggets with lightweight description, *e.g.*,

given (M, η), we sample wrt to D by sampling an observation (M) and then getting the class (η).

Bayes posterior admits the expression $\eta^\star = \pi \cdot \mathrm{dP}/\mathrm{dM}$ [25], and is optimal for *proper* losses.

Losses for class probability estimation a *loss for class probability estimation*, $\ell : \mathcal{Y} \times [0, 1] \to \mathbb{R}$, is expressed as $\ell(y, u) \doteq [\![y = 1]\!] \cdot \ell_1(u) + [\![y = -1]\!] \cdot \ell_{-1}(u)$, where $[\![.]\!]$ is Iverson's bracket [17]. Functions ℓ_1, ℓ_{-1} are called *partial* losses. A loss is *symmetric* when $\ell_1(u) = \ell_{-1}(1 - u), \forall u \in [0, 1]$ [22] and *differentiable* when both partial losses are differentiable. The pointwise conditional risk of estimator $\hat{\eta} \in [0, 1]$ with respect to ground (unknown) truth $\eta \in [0, 1]$ is:

$$L(\hat{\eta}, \eta) = \eta \cdot \ell_1(\hat{\eta}) + (1 - \eta) \cdot \ell_{-1}(\hat{\eta}). \tag{1}$$

The Bayes tilted estimate of loss ℓ [30] is the pointwise minimizer of (1), which can be set-valued:

$$t_\ell(\eta) \quad \doteq \quad \arg \inf_{u \in [0,1]} L(u, \eta). \tag{2}$$

A loss ℓ is **proper** if $\eta \in t_\ell(\eta)$ and **strictly proper** if $\{\eta\} = t_\ell(\eta)$ [25]. A loss is important for our purpose, the α-*loss*: it is a differentiable, symmetric loss defined by the following partial losses

$$\ell_1^{(\alpha)}(u) \doteq \frac{\alpha \cdot (1 - u^{\frac{\alpha-1}{\alpha}})}{\alpha - 1}, \quad \ell_{-1}^{(\alpha)}(u) \doteq \ell_1^{(\alpha)}(1 - u), \tag{3}$$

for $\alpha \geq 0$ and $\ell_1^{(\alpha)}(u) \doteq \ell_{-1}^{(-\alpha)}(u) = \ell_1^{(-\alpha)}(1 - u)$ for $\alpha < 0$. As $\alpha \to 1$, the α-loss converges to log-loss ($\ell_1^{\mathrm{LOG}}(u) \doteq -\log(u)$, strictly proper) and as $\alpha \to \infty$, it converges to the 0/1-loss ($\ell_1^{0/1}(u) \doteq [\![u < 1/2]\!]$, proper). If $\alpha \notin \{0, \infty\}$ and $\eta \neq 1/2$, its Bayes tilted estimate is [30]:

$$t_\ell(\eta) \quad = \quad \{\eta^\alpha / (\eta^\alpha + (1 - \eta)^\alpha)\}. \tag{4}$$

Importantly, for any $\eta \notin \{0, 1/2, 1\}$ and any $\eta' \in (0,1)$, there exists $\alpha \in \mathbb{R}$ in (4) such that $t_\ell(\eta) = \eta'$. This property, called *twist-properness* [30], allows to correct for any twist in η.

Population loss A model that fits a posterior η is trained to minimize a general population version of (1), called *risk*, which integrates the Bayes tilted estimate, as:

$$L(\eta; \mathrm{M}, \eta^\star) \quad \doteq \quad \mathbb{E}_{\mathsf{X}\sim\mathrm{M}}\left[L(t_\ell(\eta(\mathsf{X})), \eta^\star(\mathsf{X}))\right], \tag{5}$$

where $\eta, \eta^\star \in [0,1]^{\mathcal{X}}$ (with slight notational abuse from the pointwise case); and $\eta^\star$ denoting the true posterior (*i.e.*, approximated from data), but in general can be switched for any other posterior. If the loss is proper, such as for the log-loss, we retrieve the classical expression $L(\eta; \mathrm{M}, \eta^\star) = \mathbb{E}_{\mathsf{X}\sim\mathrm{M}}\left[L(\eta(\mathsf{X}), \eta^\star(\mathsf{X}))\right]$. Further note that taking $\eta = \eta^\star$ results in (5) simplifying to the population / Bayes risk. Note: evaluating the risk using $\eta^\star$ can simply be approximated by taking empirical expectations. We now explain how we put together the elements presented here for fairness correction.

4 Making black-boxes fair with guarantees

The overall recipe η_{u} is an accurate but eventually unfair black-box. We wish to learn a fair posterior, η_{f}, which is a function of η_{u}, but we neither "open" nor train η_{u} further. We thus want a mapping

$$\eta_{\mathrm{u}} \quad \mapsto \quad \eta_{\mathrm{f}} \tag{6}$$

with desirable analytical, representation, and algorithmic properties (*i.e.* desiderata in Section 1). We add an **(invertibility)** condition which states that η_{f} must be easy to be decomposed into its original black-box η_{u} and its other components. η_{f} integrates components that need to be learned from data to achieve fairness; as such, we need an algorithm, say $\mathcal{A}$:

$\mathcal{A}$ learns η_{f} by minimizing a risk as:

$$\eta_{\mathrm{f}} \overset{\mathcal{A}}{\leftarrow} \min L(\eta_{\mathrm{u}}; \mathrm{M}, \eta_{\mathrm{t}}), \tag{7}$$

where loss ℓ, mixture M, and "target" posterior η_{t} are *designed* to satisfy a **(flexible)** selection of fairness criteria.

Assuming that the loss is fixed, for fairness one goal could be to improve the risk of the worst effected subgroup s. In this case, one could pick M_s the mixture M conditioned on $S = s$; and refining over the data we want to predict on $\eta^\star$. Further details and specifics can be found in Section 6.

Our implementation of mapping (6) and (7) We know that the Bayes tilted estimate of the α-loss can correct any posterior to a desired value, so we are going to use (4) as our implementation of (6):

$$\eta_{\mathrm{f}}(\boldsymbol{x}) \quad \doteq \quad \frac{\eta_{\mathrm{u}}(\boldsymbol{x})^{\alpha(\boldsymbol{x})}}{\eta_{\mathrm{u}}(\boldsymbol{x})^{\alpha(\boldsymbol{x})} + (1-\eta_{\mathrm{u}}(\boldsymbol{x}))^{\alpha(\boldsymbol{x})}} \in [0,1] \tag{8}$$

where $\alpha \in \mathbb{R}_*^{\mathcal{X}}$ thus defines what needs to be learned to *wrap* η_{u}. There remains to pick the *loss* to be used in L. We choose the log-loss for two reasons: ① it is strictly proper and so solving (7) "pushes" η_{f} towards target η_{t}, ② it is also the α-loss for $\alpha = 1$ so we are guaranteed that $\eta_{\mathrm{f}} \to \eta_{\mathrm{u}}$ *is equivalent to* $\alpha \to 1$ (otherwise we might have only $\Leftarrow$). We now make this more quantitative.

(Proximity) guarantees under bounded fairness corrections Since posteriors $\eta_{\mathrm{u}}, \eta_{\mathrm{f}}$ have the same support, a good divergence measure is an f-divergence; we pick the KL divergence [2]:

$$\mathrm{KL}(\eta_{\mathrm{u}}, \eta_{\mathrm{f}}; \mathrm{M}) \quad = \quad \mathbb{E}_{(\mathsf{X},\mathsf{Y})\sim\mathrm{D}_{\mathrm{u}}}\left[\log\left(\frac{\mathrm{dD}_{\mathrm{u}}((\mathsf{X},\mathsf{Y}))}{\mathrm{dD}_{\mathrm{f}}((\mathsf{X},\mathsf{Y}))}\right)\right], \tag{9}$$

where $\mathrm{D}_{\mathrm{u}}, \mathrm{D}_{\mathrm{f}}$ are the product measures defined from M and their respective posteriors (Section 3). To bound (9), we introduce mild assumptions to the black-box η_{u} and corresponding $\alpha(.)$ correction (SI, Section II presents a more general result). The former comes as a calibration of $\eta_{\mathrm{u}}(.)$.

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

$$\mathrm{Im}(\eta_u) \subseteq \mathbb{I} \doteq \left[\frac{1}{1+\exp(B)}, \frac{1}{1+\exp(-B)}\right] \textit{ (a.s.).} \tag{10}$$

Compliance with Assumption 1 can be done by clipping the black-box's output with user-fixed B or making sure it is calibrated and then finding B. We now present setting **(S1)**.

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

This setting lead to the following bound on the distortion.

Theorem 1. ***(S1)*** *implies* $\mathrm{KL}(\eta_u, \eta_f; \mathrm{M}) \leq \pi^2/(6 \cdot (2 + \exp(B) + \exp(-B)))$.

To get a glimpes into the quality of the bound, fix $B = 3$ for **(S1)**. In this case, we want $\alpha(.) \in [2/3, 4/3]$ (a.s.), which is a reasonable sized interval centered at 1, the clamped black-box posterior's

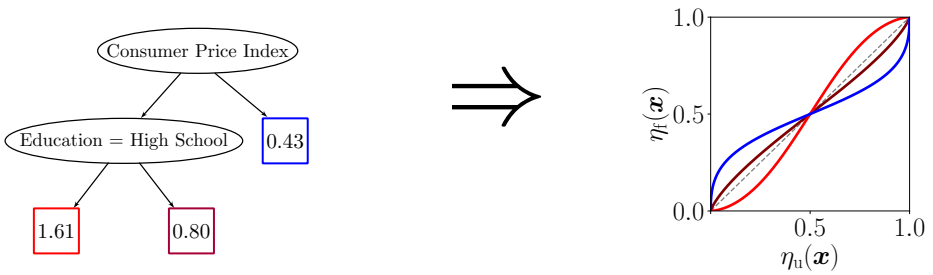

Figure 1: Left: An example (sub) α-tree trained on the Bank dataset (SI, Section XI) for CVaR. The full tree can be found in SI, Section XVIII. Right: The corresponding correction of posteriors. As per (8), different curves correspond to different partitions of the domain. Multiple curves can correspond to the same sensitive attribute modality (two red shaded curves).

interval is approximately $[0.04, 0.96]$, which is quite flexible, and the distortion is upperbounded as $\mathrm{KL}(\eta_{\mathrm{u}}, \eta_{\mathrm{f}}; \mathrm{M}) \leq 7.5E-2$. This bound, relevant to **(proximity)**, holds for *any* representation of α.

Overview of (7) to make the high-level process precise, we thus look after the minimization of

$$L(\eta_{\mathrm{u}}; \mathrm{M}, \eta_{\mathrm{t}}) \doteq \mathop{\mathbb{E}}_{\mathsf{X}\sim\mathrm{M}} \left[\eta_{\mathrm{t}}(\mathsf{X}) \cdot -\log \eta_{\mathrm{f}}(\mathsf{X}) + (1-\eta_{\mathrm{t}}(\mathsf{X})) \cdot -\log(1-\eta_{\mathrm{f}}(\mathsf{X}))\right], \tag{11}$$

with η_{f} in (8). (11) has a simple and popular alternative expression: plugging (8) in (11) and simplifying yields the expression based on the *logistic loss* (we use $\mathrm{M}, \eta_{\mathrm{t}}$ to craft D_{t}):

$$L(\eta_{\mathrm{u}}; \mathrm{M}, \eta_{\mathrm{t}}) \doteq \mathop{\mathbb{E}}_{(\mathsf{X},\mathsf{Y})\sim\mathrm{D}_{\mathrm{t}}} \left[\log\left(1+\exp\left(-\mathsf{Y}\alpha(\mathsf{X})\log\left(\frac{\eta_{\mathrm{u}}(\mathsf{X})}{1-\eta_{\mathrm{u}}(\mathsf{X})}\right)\right)\right)\right]. \tag{12}$$

Remark The analytical form in (8) also brings the following easy-to-check **(composability)** property.

Lemma 1. *The composition of any two wrapping transformations* $\eta_u \overset{\alpha}{\mapsto} \eta_f \overset{\alpha'}{\mapsto} \eta'_f$ *following* (8) *is equivalent to the single transformation* $\eta_u \overset{\alpha\cdot\alpha'}{\mapsto} \eta'_f$.

This also gives us compliance with the **(invertibility)** condition by wrapping η_{f} using $\alpha' = 1/\alpha$. We now investigate the functional form of α and how to train it with boosting-compliant convergence.

5 Alpha-trees and how to grow them

We now focus on three main components with key focus on **(convergence)** and additional leverage on **(explainability)**: the models we use for function α, a convenient upperbound on (11), and, a boosting-compliant algorithm to minimize this upperbound in learning. Our algorithm TopDown ends up being a generalization of standard boosting for decision tree classification [15]. At this stage, both the mixture M and η_{t} remain unspecified as they will depend on the fairness objective tackled.

Alpha-trees We first define the functions we use for α.

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.

Figure 1 presents an example of α-tree. Just like a decision tree, an α-tree recursively splits the whole domain $\mathcal{X}$, the difference being that leaf predictions are correction to the unfair posterior, not labels. If the splits are simple, sub-α-trees can be scrutinized in an **(explainable)** way.

General induction of an α-tree Assumption 1 is instrumental for this part. Denote $\iota(u) \doteq \log(u/(1-u))$ the logit of $u \in [0,1]$ and $\tilde{\iota}(u) \doteq \iota(u)/B$ a normalization which satisfies $\tilde{\iota}(\mathbb{I}) = [-1, 1]$ (10). Also, we define the edge of the normalized logit given M and η_{t},

$$\mathsf{e}(\mathrm{M}, \eta_{\mathrm{t}}) \doteq \mathbb{E}_{(\mathsf{X},\mathsf{Y})\sim\mathrm{D}_{\mathrm{t}}}\left[\mathsf{Y}\tilde{\iota}(\eta_{\mathrm{u}}(\mathsf{X}))\right], \tag{13}$$

which satisfies $\mathsf{e}(\mathrm{M}, \eta_{\mathrm{t}}) \in [-1, 1]$ when Assumption 1 is satisfied. The blueprint of our algorithm, TopDown, is given in Algorithm 1, where $\mathcal{H}$ denote a function set for splits, each element of which is a function from $\mathcal{X}$ to $\{-1, 1\}$, $+1$ indicating the observation follows the right arc at the split. TopDown is similar at a high level to classical top-down decision trees induction algorithms [15, Figure 1]. A notable low-level difference is the initial α-tree provided, Υ_0; using the decision tree induction blueprint would require Υ_0 to be a 1-node tree. Another difference is the loss used for selecting split(s). We now present this criterion, letting $H(q) \doteq -q\log(q) - (1-q)\log(1-q)$.

Algorithm 1 TOPDOWN $(\mathrm{M}, \eta_t, \Upsilon_0, B)$

Input mixture M, 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 $\lambda^\star \in \Lambda(\Upsilon)$
  Step 2.2 : $h^\star \leftarrow \arg\min_{h \in \mathcal{H}} \mathrm{H}(\Upsilon(\lambda^\star, h); \mathrm{M}, \eta_t)$;
  Step 2.3 : $\Upsilon \leftarrow \Upsilon(\lambda^\star, h^\star)$; // split using $h^\star$ at $\lambda^\star$
Step 3 : label leaves:

$$\Upsilon(\lambda) \doteq \tilde{\iota}\left(\frac{1 + \mathsf{e}(\mathrm{M}_\lambda, \eta_t)}{2}\right), \forall \lambda \in \Lambda(\Upsilon), \quad (15)$$

Output Υ;

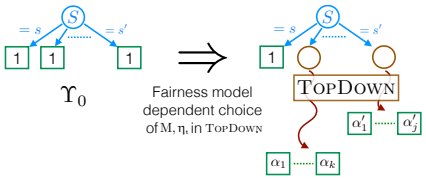

Figure 2: Picking Υ_0 a stump on the fairness attribute allows to finely tune growths of sub-α-trees to the fairness criterion at hand.

Definition 2. *Given α-tree Υ with leafset Λ, when Assumption 1 is satisfied, the **entropy of** Υ is:*

$$\mathrm{H}(\Upsilon; \mathrm{M}, \eta_t) \quad \doteq \quad \mathbb{E}_{\lambda \sim \mathrm{M}_{\Lambda(\Upsilon)}} \left[\mathrm{H}_1(\lambda; \mathrm{M}, \eta_t)\right], \quad (14)$$

where $\mathrm{H}_1(\lambda; \mathrm{M}, \eta_t) \doteq H\left((1 + \mathsf{e}(\mathrm{M}_\lambda, \eta_t))/2\right)$, M_λ is M conditioned to leaf $\lambda \in \Lambda$ and $\mathrm{M}_{\Lambda(\Upsilon)}$ is measure induced on $\Lambda(\Upsilon)$ by the leaves' weights on M.

TOPDOWN is a boosting algorithm To show that TOPDOWN is a boosting algorithm, we need a *Weak Hypothesis Assumption*, which postulates informally that each chosen split brings a small edge over random splits for a tailored distribution – *locally* making the problem "harder".

Definition 3. *Let $\lambda \in \Lambda(\Upsilon)$ and $\mathrm{D}_{t\lambda}$ be the product measure on $\mathcal{X} \times \mathcal{Y}$ conditioned on λ. The **balanced** product measure $\mathrm{D}'_{t\lambda}$ at leaf λ is defined as ($z \doteq (\boldsymbol{x}, y)$ for short):*

$$\mathrm{D}'_{t\lambda}(z) \quad \doteq \quad \frac{1 - \mathsf{e}(\mathrm{M}_\lambda, \eta_t) \cdot y\tilde{\iota}(\eta_u(\boldsymbol{x}))}{1 - \mathsf{e}(\mathrm{M}_\lambda, \eta_t)^2} \cdot \mathrm{D}_{t\lambda}(z). \quad (16)$$

We check that $\int_\lambda \mathrm{dD}'_{t\lambda} = 1$ because of the definition of $\mathsf{e}(\mathrm{M}_\lambda, \eta_t)$ (13). Our balanced distribution was named after [15]'s: ours indeed generalizes theirs. The key difference comes from the change in setting, where we not only care about the purity of a leaf but also its agreement with the unfair posterior η_u (13). The "*fairness-free case*" where $\tilde{\iota}(.)$ is replaced by constant 1 yields the original balanced distribution [15]. We now state our Weak Hypothesis Assumption (WHA).

Assumption 2. *Let $h : \mathcal{X} \to \{-1, 1\}$ be the function splitting leaf λ, and let $\gamma > 0$. We say that h γ-witnesses the Weak Hypothesis Assumption (**WHA**) at λ iff*

$$\textbf{(i)} \left| \mathop{\mathbb{E}}_{(\mathsf{X},\mathsf{Y}) \sim \mathrm{D}'_{t\lambda}} \left[\mathsf{Y}\tilde{\iota}(\eta_u(\mathsf{X})) \cdot h(\mathsf{X})\right] \right| \geq \gamma; \quad \textbf{(ii)}\ \mathsf{e}(\mathrm{M}_\lambda, \eta_t) \cdot \mathop{\mathbb{E}}_{(\mathsf{X},\mathsf{Y}) \sim \mathrm{D}_{t\lambda}} \left[(1 - \tilde{\iota}^2(\eta_u(\mathsf{X}))) \cdot h(\mathsf{X})\right] \leq 0.$$

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 distribution, if we consider the fairness-free case where we only care about the 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 X. We now state TOPDOWN's boosting compliant **(convergence)**.

Theorem 2. *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 $h^\star$ (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 $|\Lambda(\Upsilon)| \geq (1/\varepsilon)^{c \log\left(\frac{1}{\varepsilon}\right)/\gamma^2}$, then η_f crafted from (8) using* TOPDOWN*'s Υ achieves $L(\eta_u; \mathrm{M}, \eta_t) \leq \varepsilon$.*

The proof of Theorem 2 is in SI, Section V: it proceeds in two stages, the first being the proof that

$$L(\eta_u; \mathrm{M}, \eta_t) \quad \leq \quad \mathrm{H}(\Upsilon; \mathrm{M}, \eta_t), \quad (17)$$

with the scoring in (15), the second being the boosting results focused on the entropy H of the α-tree.

An audacious scoring scheme for α-trees Let us call *conservative* the scoring scheme in (15). 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.

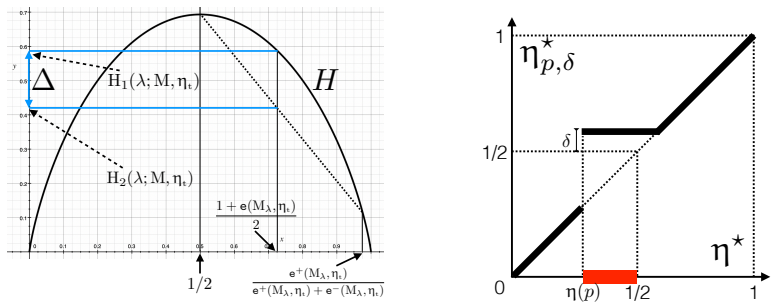

Figure 3: *Left*: Difference between the per-leaf bounds on risk $L(\eta_u; \mathrm{M}, \eta_t)$ using (14) and (17) (conservative scoring) and (20) (audacious scoring). Details in the proof of Lemma 2. *Right*: A representation of the (p, δ)-pushup of $\eta^\star$, where $\eta(p) \doteq \inf \eta^\star(\mathcal{X}_p) < 1/2$ (Def. 5). All posteriors in $[\eta(p), 1/2 + \delta]$ are mapped to $1/2 + \delta$; others do not change. New posterior $\eta^\star_{p,\delta}$ eventually reduces the accuracy of classification for observations whose posterior lands in the thick red interval (x-axis).

Definition 4. *For any mixture* M *and posteriors* η_u, η_t*, let*

$$\mathsf{e}^+(\mathrm{M}, \eta_t) \doteq \mathop{\mathbb{E}}_{(\mathsf{X},\mathsf{Y})\sim \mathrm{D}_t} [\max\{0, \mathsf{Y}\tilde{\iota}(\eta_u(\mathsf{X}))\}] \,; (18) \qquad \mathsf{e}^-(\mathrm{M}, \eta_t) \doteq -\mathop{\mathbb{E}}_{(\mathsf{X},\mathsf{Y})\sim \mathrm{D}_t} [\min\{0, \mathsf{Y}\tilde{\iota}(\eta_u(\mathsf{X}))\}] \,. \qquad (19)$$

The ***audacious*** *scoring schemes at the leaves of the* α*-tree replaces* (15) *in Step 3 by:*

$$\Upsilon(\lambda) \quad \doteq \quad \tilde{\iota}\left(\frac{\mathsf{e}^+(\mathrm{M}_\lambda, \eta_t)}{\mathsf{e}^+(\mathrm{M}_\lambda, \eta_t) + \mathsf{e}^-(\mathrm{M}_\lambda, \eta_t)}\right), \quad \forall \lambda \in \Lambda(\Upsilon).$$

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

$$\mathrm{H}_2(\lambda; \mathrm{M}, \eta_t) \doteq \log(2) \cdot \left(1 + (\mathsf{e}^+_\lambda + \mathsf{e}^-_\lambda) \cdot \left(H_2\left(\frac{\mathsf{e}^+_\lambda}{\mathsf{e}^+_\lambda + \mathsf{e}^-_\lambda}\right) - 1\right)\right),$$

where let $\mathsf{e}^b_\lambda \doteq \mathsf{e}^b(\mathrm{M}_\lambda, \eta_t), \forall b \in \{+, -\}$*. Using audacious scoring, we get instead of* (17)*:*

$$L(\eta_u; \mathrm{M}, \eta_t) \quad \leq \quad \mathbb{E}_{\lambda\sim \mathrm{M}_{\Lambda(\Upsilon)}} [\mathrm{H}_2(\lambda; \mathrm{M}, \eta_t)] \,. \qquad (20)$$

While upperbounds in (17) and (20) may look non comparable, it takes a simple argument to show that (20) is never worse and can me much tighter.

Lemma 2. $\forall \alpha$*-tree* Υ*,* $\mathbb{E}_{\lambda\sim \mathrm{M}_{\Lambda(\Upsilon)}} [\mathrm{H}_2(\lambda; \mathrm{M}, \eta_t)] \leq \mathrm{H}(\Upsilon; \mathrm{M}, \eta_t)$.

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 2. It also takes a simple picture to show that the per-leaf slack in Lemma 2 can be substantial, a slack which can be represented using a simple picture, see Figure 3 (left), following from the use of Jensen's inequality in the Lemma's proof.

Conservative vs audacious corrections If we were to just care about accuracy, we would barely have any reason to use the conservative correction. Even thinking about generalisation, the Rademacher complexity of decision trees is a function of their depth so the faster the convergence, the better [3, Section 4.1] (see also Section 7). Adding fairness substantially changes the picture: some constraints, like equality of opportunity (Section 6) can antagonise accuracy to some extent. In such a case, using the conservative correction can keep posteriors η_u and η_t close enough (Corollary 1 / Theorem A) so that fairness can be achieved without substantial sacrifice on accuracy.

A convenient initial alpha-tree Since the fairness attribute partitions the dataset, there is a simple and convenient choice for Υ_0 in TOPDOWN, the stump whose test is on the fairness attribute (thus, not necessarily binary), resulting in separate sub-α-trees for each modality. We then run TOPDOWN with a specific choice of mixture and target posterior to accommodate the fairness model at hand (Fig. 2).

6 Handling fairness notions

At this stage, the only remaining unknowns are the fairness-model dependent M, η_t (§ 5, Par. 1). We consider three different fairness criteria: conditional value at risk, equality of opportunity, and statistical parity; and provide corresponding M, η_t for each criteria. The choices made for our these fairness notions are not meant to be optimal, as other choices could provide substantial leverage, but instead provides an illustration that our approach can be adapted to a variety of criteria. Due to the

lack of space, we defer the case of statistical parity to SI (pg 28).

Conditional value at risk CVaR was introduced in optimization / finance [26] and its use in fairness for ML was introduced in [33]. The criterion to minimize is:

$$\text{CVaR}_\beta(\eta_\text{f}) \doteq \mathbb{E}_{\mathsf{S}\sim \text{M}_S}[L(\eta_\text{u}; \text{M}_\mathsf{S}, \eta^\star) | L(\eta_\text{u}; \text{M}_\mathsf{S}, \eta^\star) \geq L_\beta],$$

L_β being the risk value for the β quantile among groups, which is user defined; also, M_S is the measure induced on S by the groups' weights and M_s is the mixture conditioned on $S = s$. CVaR focuses optimization on the worst treated groups and if we denote $\mathcal{S}_\beta$ the subset of modalities used in CVaR, then a simple way to optimize CVaR is to repeatedly grow the subtree of the α-tree that makes the correction for one of those groups. Put simply, we iterate

$$\boxed{\text{TopDown with } \text{M} \leftarrow \text{M}_s \ (s \in \mathcal{S}_\beta) \text{ and } \eta_\text{t} \leftarrow \eta^\star,}$$

and we repeat until $\text{CVaR}_\beta(\eta_\text{f})$ gets below a threshold or (more specifically) its worst treated group gets a risk below a threshold (this can be used as stopping criterion). This imposes to update $\mathcal{S}_\beta$ to keep the set accurate between runs of TopDown. The number of iteration to get CVaR_β below a threshold ε in our boosting framework is thus no more than the number of modalities of S times the $|\Lambda(\Upsilon)|$ bound in Theorem 2.

Equality of opportunity (EOO) requires to smooth discrimination within an "advantaged" group, modeled by the label $y = 1$ [14]. We say that η_f achieves ε-equality of opportunity iff a mapping h_f of η_f to $\mathcal{Y}$ (*e.g.* using the sign of its logit) satisfies

$$\max_{s\in S} \mathop{\mathbb{P}}_{\mathsf{X}\sim \text{P}_s}[h_\text{f}(\mathsf{X}) = 1] - \min_{s\in S} \mathop{\mathbb{P}}_{\mathsf{X}\sim \text{P}_s}[h_\text{f}(\mathsf{X}) = 1] \leq \varepsilon, \tag{21}$$

where P_s is the positive observations' measure conditioned to value $S = s$ for the sensitive attribute. EOO can be antagonistic with the fitting of η_f to $\eta^\star$: if that latter one is close to zero in a subgroup and close to one in another one, then better fittings on η_f can arbitrarily increase the LHS in (21). To cope with this issue, we do not pick $\eta_\text{t} \leftarrow \eta^\star$ as in CVaR, but rather skew the posterior for a subset of observations. Fix some $s^\circ \in \arg\min_{s\in S} \mathbb{P}_{\mathsf{X}\sim \text{P}_s}[h_\text{f}(\mathsf{X}) = 1]$. Our strategy consists in skewing the target posterior for $S = s^\circ$ so that for a subset of the subgroup, it becomes $> 1/2$. A convenient use of TopDown then yields more positive classifications for $S = s^\circ$ – thus a more fair outcome – and thus a reduction of LHS in (21) until (21) is satisfied[1]. For this, we create a (p, δ)-*pushup* of $\eta^\star$.

Definition 5. *Fix $p \in [0, 1]$ and let $\mathcal{X}_p$ be a subset of $\mathcal{X}$ such that (i) $\inf \eta^\star(\mathcal{X}_p) \geq \sup \eta^\star(\mathcal{X} \backslash \mathcal{X}_p)$ and (ii) $\int_{\mathcal{X}_p} \text{dM} = p$. For any $\delta \geq 0$, the (p, δ)-pushup of $\eta^\star$, $\eta^\star_{p,\delta}$, is the posterior defined as $\eta^\star_{p,\delta} = \eta^\star$ if $\inf \eta^\star(\mathcal{X}_p) \geq 1/2$ and otherwise:*

$$\eta^\star_{p,\delta}(\boldsymbol{x}) \doteq \begin{cases} \eta^\star(\boldsymbol{x}) & \textit{if} \quad (\boldsymbol{x} \notin \mathcal{X}_p) \vee \left(\eta^\star(\boldsymbol{x}) \geq \frac{1}{2} + \delta\right) \\ \frac{1}{2} + \delta & \textit{otherwise.} \end{cases} \tag{22}$$

Fig. 3 (right) presents an example of this mapping. Notice that the transformation can introduce classification mistakes with respect to $\eta^\star$, but only examples with (i) small "edge" $|1/2 - \eta^\star|$ and (ii) labeled as negative on $\eta^\star$ are susceptible to get positive label on $\eta^\star_{p,\delta}$. Notice the tradeoff: (ii) is consistent with the fairness objective while (i) limits the degradation in accuracy. We then run TopDown using as mixture the *positive* measure conditioned to $S = s^\circ$ and $p \doteq \mathbb{P}_{\mathsf{X}\sim \text{P}_{s^*}}[h_\text{f}(\mathsf{X}) = 1] + \varepsilon/(K - 1), \delta \doteq K\varepsilon/(K - 1)$, with $K > 1$ user-fixed. Thus, we do

$$\boxed{\text{TopDown with } \text{M} \leftarrow \text{P}_{s^\circ} \text{ and } \eta_\text{t} \leftarrow \eta^\star_{p,\delta},}$$

and we have the following guarantee:

Theorem 4. *If* TopDown *is run until $L(\eta_f; \text{M}, \eta_t) \leq (\varepsilon^4/2) + \mathbb{E}_{\mathsf{X}\sim \text{M}}[H(\eta_t(\mathsf{X}))]$, then after the run we observe $\mathbb{P}_{\mathsf{X}\sim \text{P}_{s^\star}}[h_f(\mathsf{X}) = 1] - \mathbb{P}_{\mathsf{X}\sim \text{P}_{s^\circ}}[h_f(\mathsf{X}) = 1] \leq \varepsilon$.*

For the optimization to be carried out properly in the full context of EOO, we should not wait to get the bound on $L(\eta_\text{u}; \text{M}, \eta_\text{t})$. Rather, we should make sure (a) we update $\arg\min_{s\in S} \mathbb{P}_{\mathsf{X}\sim \text{P}_s}[h_\text{f}(\mathsf{X}) = 1]$ (and thus s°) after each split in the α-tree and (b) we keep $\arg\max_{s\in S} \mathbb{P}_{\mathsf{X}\sim \text{P}_s}[h_\text{f}(\mathsf{X}) = 1]$ 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. One should note that the guarantee presented in Theorem 4 and Section 6 depends on the mapping h_f and not the direct posterior η_f as typically

[1] A symmetric strategy holds if one instead wants to *reduce* $\mathbb{P}_{\mathsf{X}\sim \text{P}_{s^*}}[h_\text{f}(\mathsf{X}) = 1]$ ($s^* \in \arg\max_{s\in S} \mathbb{P}_{\mathsf{X}\sim \text{P}_s}[h_\text{f}(\mathsf{X}) = 1]$). Choosing one strategy depends on the application: if positive class implies money spending (*e,g,* for loan prediction), then our strategy implies spending more money to achieve fairness, while the latter one reduces the amount of money lent to achieve fairness.

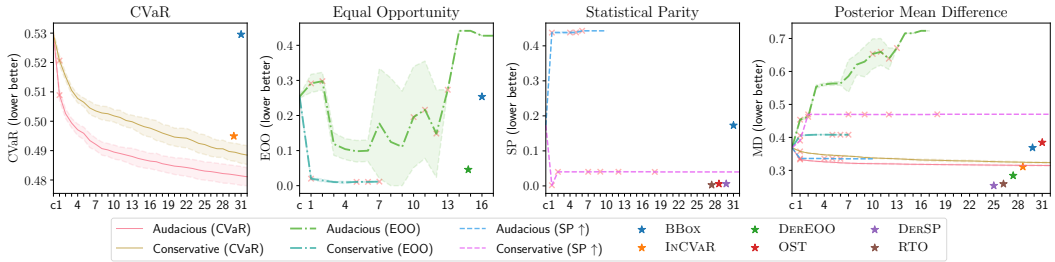

Figure 4: **ACS 2015 with Binary Sensitive Attribute and Random Forest Black-box**: Evaluation of TopDown over boosting iterations (x-axis) for different fairness criteria. 'c' on the x-axis denotes the clipped black-box. '$\times$' denote when a subgroup's α-tree is initiated (over any fold). The shade denotes $\pm$ a standard deviation from the mean, this disappears when folds have early stopping.

considered [14]. When taking the mapping as a threshold of the posterior (sign of the logit), $h_{\rm f}$ can be interpreted as forcing the original posterior to be extreme values of 0 or 1. It should be noted that (22) requires an explicit approximation of $\eta^\star$. However, if one does not want to make such an approximation, one can adapt the statistical parity approach (details in Appendix IX). Similarly, if wants to consider the typical EOO definitions depending on posterior values, the target measure can be replaced (*i.e.*, replacing the measure M with the measure of the positive examples P).

7 Discussion

Generalization As we are utilizing the logistic loss (12), a relevant capacity notion to assess the uniform convergence of this risk for the whole wrapped model is the Rademacher **(complexity)** of the following set of functions [3]:

$$\mathcal{H}_{\rm f} \doteq \left\{ \alpha(\boldsymbol{x}) \cdot \log\left(\frac{\eta_{\rm u}(\boldsymbol{x})}{1-\eta_{\rm u}(\boldsymbol{x})}\right) : \forall(\alpha, \eta_{\rm u}) \right\}, \tag{23}$$

where we assume known the set of functions from which $\eta_{\rm u}$ was trained. We now assume we have a m-training sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i) \sim \mathrm{D}\}_{i=1}^m$. The empirical Rademacher complexity of a set of functions $\mathcal{H}$ from $\mathcal{X}$ to $\mathbb{R}$, $\mathfrak{R}_{\mathcal{S}}(\mathcal{H}) \doteq \mathbb{E}_{\boldsymbol{\sigma}} \sup_{h\in\mathcal{H}} \mathbb{E}_i[\sigma_i h(\boldsymbol{x}_i)]$ (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 logistic loss. To see how the α-tree affects the Rademacher complexity of classification using $\eta_{\rm f}$ instead of $\eta_{\rm u}$, suppose real-valued prediction based on $\eta_{\rm u}$ is achieved via logit mapping, $\iota \circ \eta_{\rm u}$ (23). Such mappings are common for decision trees [27].

Lemma 3. *Suppose $\{\eta_u\}$ is the set of decision trees of depth $\leq d$ and denote $\mathfrak{R}_{\mathcal{S}}(\mathrm{DT}(d))$ 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 (23): $\mathfrak{R}_{\mathcal{S}}(\mathcal{H}_f) \leq \mathfrak{R}_{\mathcal{S}}(\mathrm{DT}(d+d'))$.*

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

Sensitive feature use vs proxy-based prediction 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 [36, § 6.2.3]. Our basic approach to the induction of α-trees falls in the category (Fig. 2), 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 Υ_0 in TopDown. 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].

8 Experiments

To evaluate TopDown, we broadly considered 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 `Folktables`[2] [9]. The SI (pg 31) 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

[2]Public at: `github.com/zykls/folktables`

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 28) 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 [24]. 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 $\eta^\star$.

Multiple fairness criteria We evaluate TOPDOWN for CVAR, equality of opportunity EOO, and statistical parity SP. We have pushed to SI (Sections IX, XII) the complete treatment of SP, that we just summarise here. SP aims to make subgroup's expected posteriors similar and is popular in a various post-processing methods [32, 1]. The definition can be found in SI (pg 28) along with the strategy used in TOPDOWN. For SP, we present one strategy where we target the largest subgroup posterior (denoted by $\uparrow$). A symmetric strategy of targeting the smallest subgroup posterior is presented in SI. 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. 2.

To evaluate TOPDOWN, we compare against 5 baseline approaches. For CVAR we consider the in-processing approach (INCVAR) presented in [33]. For EOO, we consider a derived predictor (DEREOO) [14]. Our SP baselines include an optimized score transformation approach (OST) [32]; a derived predictor modified for SP (DERSP) [14]; and a randomized threshold optimizer approach (RTO) [1]. The clipped black-box is also displayed (BBOX).

The experiments are summarized in Fig. 4. For clarity we only plot the baselines and wrappers which are directly associated to each fairness criterian. In addition, 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 '$\times$'). 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 IX, resulting in potentially harsh updates. In SI (pg 28), we discuss an alternative approach using ties with optimal transport.

9 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.

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 28 th 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] D.-E. Knuth. Two notes on notation. *The American Mathematical Monthly*, 99(5):403–422, 1992.

[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] R. Nock and F. Nielsen. On the efficient minimization of classification-calibrated surrogates. In *NIPS*21*, pages 1201–1208, 2008.

[23] F. Petersen, D. Mukherjee, Y. Sun, and M. Yurochkin. Post-processing for individual fairness. *CoRR*, abs/2110.13796, 2021.

[24] 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.

[25] M.-D. Reid and R.-C. Williamson. Information, divergence and risk for binary experiments. *JMLR*, 12:731–817, 2011.

[26] R.-T. Rockafellar and S. Uryasev. Optimisation of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000.

[27] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *9 th COLT*, pages 80–91, 1998.

[28] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *MLJ*, 37:297–336, 1999.

[29] E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in NLP. In *ACL'19*, pages 3645–3650, 2019.

[30] T. Sypherd, R. Nock, and L. Sankar. Being properly improper. In *ICML'22*, 2022.

[31] T. van Erven and P. Harremoës. Rényi divergence and kullback-leibler divergence. *IEEE Trans. IT*, 60:3797–3820, 2014.

[32] 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.

[33] Robert C. Williamson and Aditya Krishna Menon. Fairness risk measures. In *International Conference on Machine Learning*, pages 6786–6797, 2019.

[34] 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.

[35] Forest Yang, Mouhamadou Cisse, and Oluwasanmi O Koyejo. Fairness with overlapping groups; a probabilistic perspective. *Advances in Neural Information Processing Systems*, 33, 2020.

[36] 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, Line 335.
   (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 XI.
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]