DATA PREPROCESSING TO MITIGATE BIAS WITH BOOSTED FAIR MOLLIFIERS

A PREPRINT

Alexander Soen
alexander.soen@anu.edu.au

Hisham Husain
hisham.husain@anu.edu.au

Richard Nock
richard.nock@data61.csiro.au

ABSTRACT

In a recent paper, Celis *et al.* (2020) introduced a new approach to fairness that corrects the data distribution itself. The approach is computationally appealing, but its approximation guarantees with respect to the target distribution can be quite loose as they need to rely on a (typically limited) number of constraints on data-based aggregated statistics; also resulting in a fairness guarantee which can be *data dependent*.

Our paper makes use of a mathematical object recently introduced in privacy – mollifiers of distributions – and a popular approach to machine learning – boosting – to get an approach in the same lineage as Celis *et al.* but without the same impediments, including in particular, better guarantees in terms of accuracy and finer guarantees in terms of fairness. The approach involves learning the sufficient statistics of an exponential family. When the training data is tabular, the sufficient statistics can be defined by decision trees whose interpretability can provide clues on the source of (un)fairness. Experiments display the quality of the results for simulated and real-world data.

1 Introduction

It is hard to exaggerate the importance that fairness has now taken within Machine Learning (ML) [Calmon et al., 2017, Celis et al., 2020, Williamson and Menon, 2019] (and references therein). ML being a data processing field, a common upstream source of biases leading to downstream discrimination is the data itself. A recent paper has extrapolated the max-entropy (max-ent) principle to debiasing the underlying data domain distribution itself [Celis et al., 2020]. The approach, designed for binary domains, proceeds in two stages: it first modifies the data distribution to get a fair 'seed' and then finds an approximation to this seed via a max-ent problem (or equivalently, a minimiser of the KL divergence) that meets domain constraints on aggregated statistics such as marginals. The approach has computational benefits but imposes aggregates that cannot be too fine-grained. In Celis et al. [2020], aggregates are attributes' marginals: the distribution learned is therefore maximally accurate *only* if attributes are pairwise independent, which is not true – otherwise, there would be no fairness problem. Maintaining both guarantees of utility and fairness are limited by the tight constraints and user-design aspects, respectively. Additionally, finding the right trade-off between accuracy and fairness may require tedious fine-tuning.

Our contribution   can be rooted in the same lineage of debiasing approaches but exploits a new mathematical object recently introduced in the context of differential privacy [Husain et al., 2020], allowing us to get an approach to fit a fair distribution with better guarantees in terms of accuracy and fairness. Figure 1 presents this object, called a mollifier, as well as the basics of our approach to converge to a distribution both which is fair and accurate. Our algorithm comes with two regimes for fairness, a perfect 'exact' one and a weaker regime whose fairness degrades gracefully with iterations to accommodate more 'aggressive' boosting schemes. Importantly, data fairness can transfer to *prediction* fairness: we show that training a sufficiently accurate model on fair data brings guarantees in the equal opportunity model. Last but not least, the fair distribution is an exponential family whose sufficient statistics can be interpreted and shed light on the source of (un)fairness when trained over tabular data. Figure 2 shows a flagrant additional source of unfairness to sex as sensitive attribute in our COMPAS data experiments (more in Section 5).
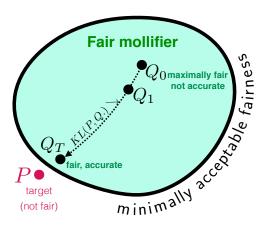
Figure 1: A mollifier $\mathcal{M}$ is a set of distributions such that every pair meets a density ratio constraint. In a fair mollifier, that we introduce, this constraint is chosen in such a way that if $\mathcal{M}$ contains a perfectly fair distribution (*e.g.*, uniform over a sensitive attribute), then *all* elements of $\mathcal{M}$ are fair. We then show how to come as close as possible within $\mathcal{M}$ to a target P – not necessarily fair –, with boosting-compliant convergence. As we relax the fairness constraint, $\mathcal{M}$ grows and ultimately contains all distributions.
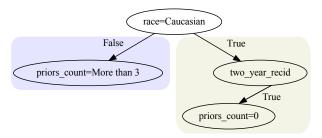

Figure 2: Decision tree classifier c_1 produced in the first round of boosting on COMPAS domain (sensitive attribute=sex, leaves not shown). A tree's most discriminative feature appears at its root: in this case, it flags a substantial related source of unfairness on the also sensitive race attribute for guessing between fair (learned) and true (not fair) distributions.

Related work   Other methods which aim to instead re-label or re-weight datasets include Kamiran et al. [2012], Kamiran and Calders [2012], Calders et al. [2009], King and Zeng [2001]. Although many of these approaches are computationally efficient, they often do not consider elements in the domain which do not appear in the dataset and lack fairness guarantees. Similarly, repair methods aim to change the input data to break the dependence on sensitive attributes for trained classifiers [Johndrow et al., 2019, Feldman et al., 2015]. To achieve distributional repair, one can employ frameworks of optimal transport [Gordaliza et al., 2019] and counterfactual distributions [Wang et al., 2019].

2   Fair Mollifiers

Representation Rate Fairness $\mathcal{D}(\mathcal{X}')$ denotes a set of distributions with common support $\mathcal{X}'$. Let $P \in \mathcal{D}(\mathcal{X} \times \mathcal{A})$, with support split into sensitive attribute $\mathcal{A}$ and non-sensitive attributes $\mathcal{X}$. Given $\tau \in (0, 1]$, a distribution P is τ-*representation rate fair* for $\mathcal{A}$ if we have

$$\frac{p[A = a_i]}{p[A = a_j]} \geq \tau, \forall a_i, a_j \in \mathcal{A}, \tag{1}$$

where $p[A = a]$ denotes the sensitive attribute's marginal distribution of P [Celis et al., 2020]. The larger τ, the more fair P. Representation rate fairness can be related to several other notions [Celis et al., 2020, Williamson and Menon, 2019], see Section 4. Hereafter, we let

$$\mathrm{RR}(P, a_i, a_j) := \frac{p[A = a_i]}{p[A = a_j]} \tag{2}$$

and the representation rate of a distribution as $\mathrm{RR}(P) := \min_{a_i, a_j \in \mathcal{A}} \mathrm{RR}(P, a_i, a_j)$.

2

Fair Mollifiers We learn fair distributions in a *mollifier*, a set with pairwise fairness-based constraint among elements.

Definition 1. Let $\mathcal{M} \subset \mathcal{D}(\mathcal{X} \times \mathcal{A})$ and $\varepsilon > 0$. $\mathcal{M}$ is an ε-**fair mollifier** iff[1] $\forall Q, Q' \in \mathcal{M}, \forall a_i, a_j \in \mathcal{A}$,

$$\text{RR}(Q, a_i, a_j) \leq \exp(\varepsilon) \cdot \text{RR}(Q', a_i, a_j).$$

Notation of Definition 1 follows Husain et al. [2020]. Hence, if $\exists F \in \mathcal{M}$ with representation rate 1, *all* distributions in $\mathcal{M}$ have a guaranteed representation rate, as follows.

Lemma 1. Suppose that $\mathcal{M}$ is an ε-fair mollifier. If there exists $F \in \mathcal{M}$ with representation rate 1, then all $Q \in \mathcal{M}$ has representation rate $\text{RR}(Q) \geq \tau$, where $\tau = \exp(-\varepsilon)$.

Mollifiers were introduced for private sampling in Husain et al. [2020]. In the case of fairness, we get the leverage that any element of the mollifier is fair. Such a property would be violated for privacy as only *sampling* is private in a mollifier. This difference is everything but anecdotical for fairness as the knowledge of the Q_T can lead to meaningful interpretations of the sources of (un)fairness as learned by the model (see Section 5, Figure 2). Given that the fairness of the distributions in the mollifier is contingent on having a fair element, we consider the *relative* mollifier construction. We first start with a reference distribution Q_0 which has representation rate 1. Then, we define the ε-fair mollifier $\mathcal{M}_{\varepsilon,Q_0}$ as the set of distributions Q satisfying $\forall a_i, a_j \in \mathcal{A}$,

$$\max \left\{ \frac{\text{RR}(Q, a_i, a_j)}{\text{RR}(Q_0, a_i, a_j)}, \frac{\text{RR}(Q_0, a_i, a_j)}{\text{RR}(Q, a_i, a_j)} \right\} \leq \exp(\varepsilon/2) \tag{3}$$

Similarly to the locally differential private mollifiers, verifying that $\mathcal{M}_{\varepsilon,Q_0}$ is a ε-fair mollifier comes from noting that for $Q, Q' \in \mathcal{M}_{\varepsilon,Q_0}$, we have

$$\frac{\text{RR}(Q, a_i, a_j)}{\text{RR}(Q', a_i, a_j)} = \frac{\text{RR}(Q, a_i, a_j)}{\text{RR}(Q_0, a_i, a_j)} \frac{\text{RR}(Q_0, a_i, a_j)}{\text{RR}(Q', a_i, a_j)} \leq \exp(\varepsilon).$$

From Eq. (3), an interesting result of having Q_0 as a perfectly fair distribution with respect to representation rate is that $\mathcal{M}_{\varepsilon,Q_0}$ will contain all $\tau = \exp(-\varepsilon/2)$ representation rate fair distributions.

Lemma 2. Suppose that $\mathcal{M}_{\varepsilon,Q_0}$ is a relative mollifier with $\text{RR}(Q_0) = 1$. If $Q \in \mathcal{D}(\mathcal{X} \times \mathcal{A})$ has representation rate $\text{RR}(Q) \geq \tau$, then $Q \in \mathcal{M}_{\varepsilon,Q_0}$, where $\tau = \exp(-\varepsilon/2)$.

Lemma 2 states that regardless of the perfectly fair Q_0 distribution, the relative mollifier $\mathcal{M}_{\varepsilon,Q_0}$ will contain all distributions with representation rate at least $\exp(-\varepsilon/2)$. The choice of reference distribution only influences the distributions with representation rate between $\exp(-\varepsilon)$ and $\exp(-\varepsilon/2)$ contained in the mollifier. This construction of relative mollifiers does not impose a choice in the elements, even when there are particular ones particularly convenient to accomodate for 'mollification'.

Fair Mollification The mollification of a distribution $P \in \mathcal{D}(\mathcal{X} \times \mathcal{A})$ is the process of finding a distribution $\hat{P}$ which minimises the KL divergence in a mollifier $\mathcal{M}$

$$\hat{P} \in \arg \min_{Q \in \mathcal{M}} \text{KL}(P, Q). \tag{4}$$

We pick the KL divergence because if is the canonical distortion for exponential families, which we use for our mollifiers. In the setting of fairness, if we only want to consider distributions with representation rate at least $\tau \in (0, 1]$, ideally, we would want to project P into the *complete* relative mollifier $\mathcal{M} = \mathcal{M}_{2\varepsilon,Q_0}$ with reference distribution $\text{RR}(Q_0) = 1$ and $\varepsilon = -\log \tau$. Practically, it is ok to rather focus on a subset of a mollifier. In our case, we will consider the incomplete mollifier $\mathcal{M}_t^{\exp} \subset \mathcal{M}_{2\varepsilon,Q_0}$ which consists of 'boosting'-able exponential families.

3 Boosting in a Fair Mollifier

Our approach to learning a fair distribution is a boosting algorithm which learns an explicit distribution with fairness guarantees and approximation guarantees for the input distribution P. We refer to the algorithm as Fair Boosted Density Estimation (FBDE); with pseudo-code in Algorithm 1. Note that the algorithm comes with several 'free' parameters, including the leveraging coefficients' scheme f.

We consider binary classifiers $c : \mathcal{X} \to \mathbb{R}$. The predicted class is denoted by $\text{sign}(c(x)) \in \{-1, 1\}$. The class our models predict is related to distinguishing between the mollifier's distribution and the target distribution. It is not 'carved' in the training sample's features and thereby does not carry the same risk for unfairness as in Pedreshi et al. [2008]. We assume boundedness on the output of c: $c(x) \in [-C, C]$ for some $C > 0$. We also require a variant of the all important *weak learning assumption* of boosting.

---

[1]Absolute continuity also required for all $\{P\} \cup \mathcal{M}$ wrt all $\mathcal{M}$.

Algorithm 1 FBDE(WL, T, τ, q_0, f)

1: **input**: Weak learner WL, # boosting iterations T,
      representation rate τ, initial conditional distribution
      $q_0(x \mid a)$ for all $a \in \mathcal{A}$, input distribution P,
      leveraging update function $f : \mathbb{N} \times [0,1] \to \mathbb{R}$;
2: $Q_0(x, a) \leftarrow q_0(x \mid a) \cdot \text{UNIF}(a)$
3: **for** $t = 1, \ldots, T$ **do**
4:     $\vartheta_t \leftarrow f(t, \tau)$
5:     $c_t \leftarrow \text{WL}(P, Q_{t-1})$
6:     $Q_t \propto Q_{t-1} \cdot \exp(\vartheta_t \cdot c_t)$
7: **end for**
8: **return**: Q_T

Definition 2 (WLA). A learner WL$(\cdot, \cdot)$ satisfies the **weak learning assumption** (WLA) for $\gamma_P, \gamma_Q \in (0, 1]$ iff for all $P, Q \in \mathcal{D}(\mathcal{X})$, WL$(P, Q)$ produces a classifier $c : \mathcal{X} \to \mathbb{R}$ satisfying $\mathbb{E}_P[c] > C \cdot \gamma_P$ and $\mathbb{E}_Q[-c] > C \cdot \gamma_Q$.

FBDE bears some similarities with the private mollified boosted density estimation algorithm [Husain et al., 2020] in that it relies on learning the sufficient statistics of an exponential family to approximate a target while meeting a constraint (fairness in our case). Technical differences include (1) the initial distribution Q_0, set to be a distribution with representation rate 1 and (2) the possibility to tune the leveraging schemes ϑ_t, to comply with specific fairness guarantees. We detail some crucial steps of FBDE.

Step 2 – Initial distribution The initial distribution Q_0 chosen needs to be perfectly fair with respect to representation rate. Thus, we restrict the marginal distribution to be uniform whilst letting the conditional distributions be free to be picked as needed. That is

$$Q_0(x, a) = q_0(x | A = a) \cdot \text{UNIF}(a), \tag{5}$$

where $\text{UNIF}(a)$ is the uniform distribution over sensitive attributes $\mathcal{A}$ and $q_0(x \mid a)$ is a user specified conditional distribution. For a finite set $\mathcal{A}$, $\text{UNIF}(a) = 1/|A|$. Practically, each conditional distribution $q_0(x | A = a)$ can be chosen as the empirical distribution of the input distribution, where samples are partitioned with respect to sensitive attributes. If a continuous conditional distribution is desirable, a fitted Gaussian distribution can be used.

Step 6 – Update equation The update equation of the boosting algorithm is given by the aggregation of the classifiers c_t given by the weak learner with weights determined by leveraging scheme ϑ_t

$$Q_t(x, a) = \frac{1}{Z_t} \exp(\vartheta_t c_t(x)) Q_{t-1}(x, a), \tag{6}$$

where Z_t is the normaliser given by

$$\begin{aligned} Z_t &= \int_{\mathcal{X} \times \mathcal{A}} \exp(\vartheta_t c_t(x)) Q_{t-1}(x, a) d(x, a) \\ &= \int_{\mathcal{A}} q_{t-1}(a) Z_t(a) da, \end{aligned} \tag{7}$$

and sensitive attribute normaliser $Z_t(a)$ is given by

$$Z_t(a) = \int_{\mathcal{X}} \exp(\vartheta_t c_t(x)) q_{t-1}(x \mid a) dx. \tag{8}$$

Here $q_k(x \mid a)$ denotes the conditional distribution after k applications of the update Eq. (6). Similarly, we denote the corresponding marginal distribution as $q_k(a)$. FBDE comes with two technical conveniences, to compute expectations and fairness guarantees.

Efficient computation of expectations: a particular convenience of exponential families, as in Eq. (6), is that computing the expectation of any function g defined over the support of Q_t can be done by *sampling* Q_0. Observe

Table 1: Summary of results for different leveraging schemes f in FBDE. Results from the 'exact' fairness rate are independent to the number of boosting iterations, unlike the 'relative' counterparts.

FAIRNESS	$f(t, \tau)$	$\mathrm{RR}(Q_t)$	$\mathcal{M}_{2\varepsilon_t, Q_0}$ SIZE ε_t
EXACT	$-\frac{\log \tau}{C 2^{t+1}}$	τ	$-\log \tau$
RELATIVE	$-\frac{\log \tau}{2Ct}$	$\tau^{O(\log t)}$	$-(1 + \log t) \log \tau$

indeed:

$$\mathbb{E}_{Q_t}[g(x, a)] = \int_{\mathcal{X} \times \mathcal{A}} \prod_{k=1}^{t} \frac{\exp(\vartheta_k c_k(x))}{Z_k} g(x, a) dQ_0$$

$$= \mathbb{E}_{Q_0} \left[\prod_{k=1}^{t} \frac{\exp(\vartheta_k c_k(x))}{Z_k} g(x, a) \right]. \tag{9}$$

Sampling from Q_0 can be significantly easier than sampling from Q_t when approximating the expectation. The same convenience applies for conditional expectations over $q_t(x \mid a)$ (details in Appendix). Efficient evaluation of the aforementioned expectations are important in both fairness and training, *i.e.*, evaluating subgroup risks Williamson and Menon [2019] and the cross entropy losses for c_t.

A simplifying trick for fairness: using the normalisation terms expressed in Eq. (7) and Eq. (8), the sensitive attribute marginal distribution $q_t(a)$ can be expressed solely normalisation terms and the initial marginal distribution

$$q_t(a) = \frac{q_{t-1}(a) Z_t(a)}{\int_{\mathcal{A}} q_{t-1}(a') Z_t(a') da'} = q_{t-1}(a) \frac{Z_t(a)}{Z_t}$$

$$= q_0(a) \prod_{k=1}^{t} \frac{Z_k(a)}{Z_k}, \tag{10}$$

where $q_0(a_i) = q_0(a_j)$ for all $a_i, a_j \in \mathcal{A}$ as per the definition of the initial distribution. Thus, the representation rate condition can be conveniently defined using the sensitive attribute normalisers given our choice of initial distribution and exponential density update:

$$\mathrm{RR}(Q_t, a_i, a_j) = \frac{q_t(a_i)}{q_t(a_j)} = \prod_{k=1}^{t} \frac{Z_k(a_i)}{Z_k(a_j)}$$

$$= \exp \left[\sum_{k=1}^{t} \log Z_k(a_i) - \log Z_k(a_j) \right]. \tag{11}$$

Using the representation rate given by Eq. (11), the leveraging scheme $f : \mathbb{N} \times [0, 1] \to \mathbb{R}$ in FBDE can be tuned to accommodate different guarantees for fairness, the first of which is 'exact' fairness.

Exact fairness guarantees: this relates to fairness that holds regardless of all other parameters in FBDE.

Theorem 3 (Exact Fairness). Suppose that $\vartheta_t := \vartheta_t^E = -\frac{1}{C 2^{t+1}} \log \tau$. Then $\mathrm{RR}(Q_t) > \tau$.

This setting is not just practical for the absolute fairness it provides: the leveraging scheme gives exponentially decreasing ϑ_t, which fits to settings for which a few classifiers are enough to break the weak learning assumption. This turns out to be observed in practice for some domains, especially when using decision trees in the weak learner. Notice that τ fairness holds regardless of the number of boosting iterations – one could boost forever and keep the fairness guarantee. Forever boosting never happening in practice, thus it is interesting to consider 'relative' fairness guarantees, where the fairness guarantee becomes weaker 'relative' to an initial fairness constraint over the boosting iterations.

Relative fairness guarantees: we compute a leveraging scheme for ϑ_t with guarantees on fairness that degrade gracefully with the number of boosting iterations.

Theorem 4 (Relative Fairness). Suppose that $\vartheta_t := \vartheta_t^R = -\frac{1}{2Ct} \log \tau$. Then $\mathrm{RR}(Q_T) > \tau^{1+\log T} = \tau^{O(\log T)}$.

Notice the key boosting difference with Theorem 3: the sum of the series of leveraging coefficients diverges, so relative fairness accommodates for more aggressive boosting schemes. Table 1 summaries the implications of the two theorems. It is not surprising that both leveraging schemes display $\vartheta_t \to 0$ as $\tau \to 1$, as maximal fairness forces the distribution to stick to Q_0 in the mollifier and is therefore data oblivious. Differences are apparent when we consider the number of boosting iterations T: should we boost for just $T = 5$, we still get a representation rate at least $\tau^{2.3}$ with relative fairness, which can still be reasonable depending on the problem. In some cases (see *e.g.* Figure 5), this is sufficient to almost reach boosting convergence and still guarantees an actual representation rate of $\tau' \approx 0.78 \approx 0.9^{2.3}$ for $\tau = 0.9$. We can also differentiate the two boosting schemes by the 'size' of the mollifiers containing the boosted densities, quantified by $\varepsilon_{\cdot}$ in Definition 1. As we see in Table 1, exact fairness guarantees a fixed mollifier size as $\varepsilon_t = -\log\tau$, while relative fairness implies a growing mollifier, for which $\varepsilon_t = -(1 + \log t)\log\tau$ instead.

Convergence guarantees for $Q_{\cdot} \to P$ Although Theorem 3 and 4 provides a guarantee on the fairness of distributions at each iteration, we have yet to show how convergence guarantees for $Q_{\cdot}$ towards P. If we set $C = \log 2$, we immediately get for both ϑ_t^E or ϑ_t^R the following Theorem.

Theorem 5. Suppose that $C = \log 2$ and $\tau > e^{-1}$. If WL satisfies the WLA for γ_P^t, γ_Q^t for $t \geq 1$. Then:

$$\mathrm{KL}(P, Q_t) \leq \mathrm{KL}(P, Q_{t-1}) - \vartheta_t \cdot \Lambda_t, \tag{12}$$

where, letting $\Gamma(z) := \log(4/(5 - 3z))$,

$$\Lambda_t = \begin{cases} \gamma_P^t \log 2 + \Gamma(\gamma_Q^t) & \text{if } \gamma_Q^t \in [1/3, 1] \text{ (HBS)} \\ \gamma_P^t + \gamma_Q^t - \frac{\log 2 \cdot \vartheta_t}{2} & \text{if } \gamma_Q^t \in (0, 1/3] \text{ (LBS)}. \end{cases} \tag{13}$$

Where HBS denotes *high* boosting regime and LBS denotes *low* boosting regime.

The proof of this theorem follows the steps of Husain et al. [2020, Theorem 5], so we inherit the convergence result, but with a catch, the way the representation rate τ parameter interacts with the update. In the HBS, we are guaranteed a positive drop in KL divergence. However in the LBS, we run into a condition mirrored in the privacy analysis of Husain et al. [2020]: we need $\gamma_P^t + \gamma_Q^t \geq \log 2 \cdot \vartheta_t/2$ for a > 0 drop in KL divergence. With the exact fairness leveraging scheme $\vartheta_t^E \propto 2^{-t} \to 0$ exponentially fast so the LBS condition vanishes rapidly. For relative fairness, we 'only' have $\vartheta_t^R \propto 1/t \to 0$ so the LBS condition holds, at least for the early boosting iterations, if the weak classifiers are not 'too weak'. Notably, each of the rates do not depend on the fairness parameter τ – a significant difference with the privacy analysis Husain et al. [2020].

In addition to the drop in KL divergence, we compute 'how far from Q_0' we progressively get, in an information-theoretic sense. For this, we let $\Delta(Q) = \mathrm{KL}(P, Q_0) - \mathrm{KL}(P, Q)$. For simplicity, we fix γ_P, γ_Q throughout the boosting procedure and assume that we are within HBS to characterise the statistical difference of densities produced by our boosting algorithm, $\mathcal{M}^{\mathrm{exp}}$ and $\widetilde{\mathcal{M}}^{\mathrm{exp}}$, where the tilda refers to the relative fairness scheme.

Theorem 6. Suppose that $C = \log 2$, $\tau > e^{-1}$, and FBDE is in HBS. Let $\alpha(\gamma) = \Gamma(\gamma)/(\gamma \log 2)$.

If $\vartheta_t := \vartheta_t^E$, then $\Delta(Q_T) \leq -\log\tau$ for $Q_T \in \mathcal{M}^{\mathrm{exp}}$ and $\forall T > 1$,

$$\Delta(Q_T) \geq -\log\tau \cdot \left\{ \frac{\gamma_P + \gamma_Q \cdot \alpha(\gamma_Q)}{2} \cdot \left(1 - \frac{1}{2^{T-1}} \right) \right\}. \tag{14}$$

If $\vartheta_t := \vartheta_t^R$, then $\Delta(\widetilde{Q}_T) \leq -(1 + \log T)\log\tau$ for $\widetilde{Q}_T \in \widetilde{\mathcal{M}}_T^{\mathrm{exp}}$ and $\forall T > 1$,

$$\Delta(\widetilde{Q}_T) \geq -\log\tau \cdot \left\{ \frac{\gamma_P + \gamma_Q \cdot \alpha(\gamma_Q)}{2} \cdot \log T \right\}. \tag{15}$$

With the exact fairness leveraging scheme ϑ_t^E, we observe that as $\gamma_P \to 1$, $\gamma_Q \to 1$, and $T \to \infty$ we have $\Delta(Q_T) = \Omega(-\log\tau)$, matching the upperbound and guaranteeing Q_T comes as close as possible to P from within the mollifier. On the other hand with the relative fairness leveraging scheme ϑ_t^R, the upperbound grows logarithmically with the boosting iterations. Additionally, there is an $\Omega(-\log\tau)$ gap between the lower and upper bound of $\Delta(\widetilde{Q}_T)$ (obtained by setting $\gamma_P = 1$ and $\gamma_Q = 1$).

4   Discussion

We put our theory in context: how our model of data fairness can imply *model* fairness and other data fairness guarantees, and how our approach compares to Celis et al. [2020].

Data fairness implies prediction fairness   A tantalising question that data fairness models face – and not just ours – is how does data fairness brings model fairness downstream a ML pipeline. This is an important question as fairness in prediction involves a model and is therefore of different nature. An useful data fairness guarantee should bring prediction fairness under ordinary constraints on model induction, that is, we should ideally have the implication:

$$\text{Fair data + `Accurate' model} \Rightarrow \text{Fair prediction.}$$

A most welcomed convenience in this case is that fair prediction relies on general purpose ML algorithms that can be chosen from a pool arguably much larger than if it were restricted to ML algorithms *specifically* designed to be fair. Based on the example of a popular notion of fairness in prediction known as *equal opportunity* Hardt et al. [2016], we show how such an implication does indeed hold from representation rate fairness. We let Y denote a target class, $\hat{Y}$ the predicted class, and A is still our sensitive attribute. For the sake of simplicity, all those attributes have binary domains, *e.g.*, Y is an admission to college variable, A is a race-related attribute, etc. With $\rho > 0$ being fixed, we say that $\hat{Y}$ satisfies ρ-equal opportunity with respect to Y, A if

$$\frac{p[\hat{Y} = 1 | A = a_i, Y = 1]}{p[\hat{Y} = 1 | A = a_j, Y = 1]} \quad \geq \quad \rho, \forall a_i, a_j \in \mathcal{A}. \tag{16}$$

The original definition in Hardt et al. [2016] holds for $\rho = 1$. A tunable approximate fairness is preferable because otherwise it is just an ideal statement of the world and it authorises tradeoffs with accuracy [Williamson and Menon, 2019], We let $\text{FNR}(\hat{Y}) = p[\hat{Y} = 0 | Y = 1]$ denote the False Negative Rate of prediction $\hat{Y}$ Menon et al. [2013].

Lemma 7. Suppose Q is τ-representation rate fair with respect to A. For any $0 \leq \rho \leq \tau$, suppose prediction $\hat{Y}$ has:

$$\text{FNR}(\hat{Y}) \quad \leq \quad \frac{\tau - \rho}{1 + \tau}. \tag{17}$$

Then $\hat{Y}$ satisfies ρ-equal opportunity with respect to Y, A.

Lemma 7 focuses on the FNR, more specific than the empirical risk but which can be controlled through a variety of criteria that correct class imbalance Menon et al. [2013]. We also notice the data-fairness friendly nature of the Lemma: as τ increases, constraints on the model error are relaxed and guarantees on equal opportunity can be tighter.

Representation rate fairness implies other data fairness   Numerous data fairness measures have been studied [Celis et al., 2020]. Without impacting all of them, a good data fairness measure should bring side guarantees on related models:

$$\text{Representation rate fair data} \Rightarrow \text{Other data fairness.}$$

One such alternative measure is *statistical rate* fairness [Celis et al., 2020] when the probability distribution domain can be further split in to include target class $Y \in \mathcal{Y}$. The statistical rate fairness constraint can be expressed by replacing the marginal distributions in Eq. (1) with conditional $p[Y = y \mid A = a]$, where $y \in \mathcal{Y}$ is fixed beforehand:

$$\frac{p[Y = y \mid A = a_i]}{p[Y = y \mid A = a_j]} \geq \rho, \tag{18}$$

where ρ is the statistical rate constant. An interesting aspect of representation rate fairness is that it allows us to control the statistical rate fairness, provided it holds on $\mathcal{Y} \times \mathcal{A}$.

Lemma 8. Suppose $P \in \mathcal{D}(\mathcal{X} \times \mathcal{Y} \times \mathcal{A})$ has representation rate τ for $\mathcal{Y} \times \mathcal{A}$. Then P has statistical rate $\rho = \tau^2$.

Another approximate measure of fairness, inspired by the '80% rule', is the *discrimination control* measure of Calmon et al. [2017], where instead of Eq. (18) we consider constraint:

$$\left| \frac{p[Y = y \mid A = a_i]}{p[Y = y \mid A = a_j]} - 1 \right| \leq \rho, \tag{19}$$

and we trivially get from Lemma 8 the following Lemma.

Lemma 9. Suppose $P \in \mathcal{D}(\mathcal{X} \times \mathcal{Y} \times \mathcal{A})$ has representation rate τ for the pair of features $(y, a) \in \mathcal{Y} \times \mathcal{A}$. Then P has discrimination control for $\rho = (1 - \tau^2)/\tau^2$.

In Calmon et al. [2017], a second measure of discrimination control is proposed, which replaces the denominator in Eq. (19) by $p[Y = y]$ and aims at making sure that the distribution of a target Y given a sensitive attribute almost
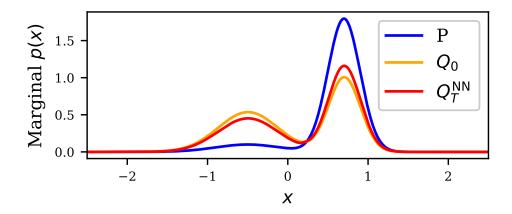
Figure 3: A comparison of the marginals of the initial distribution Q_0, the final boosted distribution Q_T, and the simulated Gaussian mixtures P. The boosting parameters are $\tau = 0.7$ and $T = 10$.

matches the target's. It is not hard to check from the proof of Lemma 8 that using the trivial fact $p[Y = y] \leq \max_i p[Y = y | A = a_i]$, representation rate τ for $P \in \mathcal{D}(\mathcal{X} \times \mathcal{Y} \times \mathcal{A})$ implies the same discrimination control as in Lemma 9. Such simple tricks would allow to show that the control of the representation rate brings guarantees on additional fairness measures, such as the maximal deviation between subgroups [Williamson and Menon, 2019, Section 2.3]. Importantly, the bounds are not data-dependent, unlike Celis et al. [2020].

Comparison with Celis et al. [2020]   The fact that our fairness guarantees are not data dependent, unlike Celis et al. [2020, Theorem 4.5], shows a major difference between approaches. It is not the most important one. With respect to ensuring fairness while remaining as close as possible to the empirical data, the max-ent approach of Celis et al. [2020] comes with downsides: optimal guarantees are on *marginals*, in small numbers for tractability (typically over single attributes) *and* one needs to hardcode the ones chosen to ensure the solution complies with fairness guarantees. This makes three key limitations, only one of which we do share. We are guaranteed to get a fairness compliant solution, provably converging to the best approximation with respect to the KL divergence, but there is also a complexity lever in our case – though arguably simpler to manoeuvre than for Celis et al. [2020]. Simply put, convergence is guaranteed *as long as* the weak learning assumption holds (Definition 2). This implies that we should not choose models that are too simple to build the sequence of sufficient statistics $c_.(\cdot)$ in the exponential family, Eq. (6) , which approximates P.

5   Experiments

Datasets   We consider two standard datasets used in the fairness ML literature and an additional synthetic dataset. Further details are presented in the Appendix.

(a) **COMPAS [Angwin et al., 2016]** contains information regarding criminal defendants in the Broward County from 2013 to 2014 [Larson et al., 2016]. We utilise the preprocessed dataset given by Bellamy et al. [2018].

(b) **Adult [Dua and Karra Taniskidou, 2017]** presents demographic information from a 1994 census, with a prediction task aimed at determining whether a person makes over 50K a year. Similar to the COMPAS, we use a preprocessed instance of the dataset from Bellamy et al. [2018].

(c) **Simulated Gaussian Mixtures** are used to test FBDE over continuous domains. Concretely, we sample values $(x, a) \in \mathbb{R} \times \{0, 1\}$ using

$$\begin{aligned} a &\sim \text{BERNOULLI}(s) \\ x &\sim \mathcal{N}(\mu_a, \sigma_a), \end{aligned} \tag{20}$$

where s determines the mixture balance between Gaussians indexed by $a \in \{0, 1\}$ with mean μ_a and standard deviation σ_a. We consider parameters $\mu_1 = -0.5$, $\mu_2 = 0.7$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, and $s = 0.9$, with 5,000 sampled points to make the dataset. Figure 3 depicts the miss-match between the initial distribution before boosting.

Architectures   On tabular data and for the weak learners of FBDE , we fit a decision tree (DT), with Gini entropy as splitting criterion and a maximum depth of 8. We denote the corresponding boosted densities using the exact fairness

Table 2: Summary of COMPAS and Adult experiments where $T = 50$ for Q_T^{DT} and Q_T^{ALT}, and $T = 10$ for Q_T^{NN}. The mean of the measurements are report across all folds and repetitions. Standard deviation values are reported in the Appendix. The representation rate of the raw data is calculated over the entire dataset.

			RAW DATA	INITIAL Q_0	BOOSTED DISTRIBUTIONS Q_T^{DT} (0.7)	Q_T^{DT} (0.9)	Q_T^{ALT} (0.7)	Q_T^{ALT} (0.9)	Q_T^{NN} (0.7)	Q_T^{NN} (0.9)	BASELINES M.ENT θ^w	M.ENT θ^b
COMPAS	SEX	RR	0.243	1.000	0.952	0.986	0.926	0.943	0.976	0.992	0.983	0.985
		SR	0.728	0.747	0.747	0.737	0.745	0.746	0.758	0.750	0.984	0.916
		$\text{KL}_{(\text{TRAIN})}$	-	0.226	0.191	0.196	0.188	0.190	0.214	0.222	0.385	0.407
		$\text{KL}_{(\text{TEST})}$	-	0.298	0.281	0.286	0.277	0.280	0.287	0.295	0.940	0.987
		$\text{TIME}_{(\text{MIN})}$	-	-	4.285	3.439	3.916	3.634	19.451	21.561	0.006	0.007
	RACE	RR	0.662	1.000	0.966	0.983	0.959	0.963	0.988	0.996	0.977	0.978
		SR	0.747	0.536	0.486	0.494	0.482	0.483	0.530	0.534	0.978	0.852
		$\text{KL}_{(\text{TRAIN})}$	-	0.048	0.020	0.020	0.019	0.019	0.042	0.046	0.140	0.139
		$\text{KL}_{(\text{TEST})}$	-	0.121	0.110	0.110	0.110	0.109	0.115	0.119	0.464	0.452
		$\text{TIME}_{(\text{MIN})}$	-	-	3.775	3.806	3.492	3.428	21.575	20.630	0.011	0.006
ADULT	SEX	RR	0.496	1.000	0.949	0.979	0.939	0.946	0.987	0.996	0.987	0.987
		SR	0.360	0.382	0.378	0.367	0.379	0.377	0.383	0.382	0.982	0.873
		$\text{KL}_{(\text{TRAIN})}$		0.061	0.054	0.055	0.053	0.053	0.059	0.060	0.350	0.696
		$\text{KL}_{(\text{TEST})}$		0.087	0.089	0.090	0.088	0.088	0.085	0.086	0.483	0.901
		$\text{TIME}_{(\text{MIN})}$	-	-	34.910	23.517	37.564	38.315	121.645	112.103	0.013	0.009

ϑ_t^E and relative fairness ϑ_t^R leveraging scheme by Q_t^{DT} and Q_t^{ALT}, respectively. We also consider neural network classifiers using the exact leveraging scheme ϑ_t^E, denoted as Q_t^{NN}, with architecture:

$$\mathcal{X} \times \mathcal{A} \xrightarrow[\text{dense}]{\text{ReLU}} \mathbb{R}^{20} \xrightarrow[\text{dense}]{\text{ReLU}} \mathbb{R}^{20} \xrightarrow[\text{dense}]{\text{sigmoid}} (0,1), \tag{21}$$

where $\mathcal{A} = \{0, 1\}$ for each experiment and $\mathcal{X}$ depends on the specific dataset we are training on. The c_t is trained using the cross entropy loss function, with Eq. (9). Further neural network details are presented in the Appendix. Over all experiments, we use 5-fold cross validation in evaluation. For the DT variants, we boost for up to $T = 50$ iterations in the real-world datasets. For the neural network classifier densities and synthetic experiments, we boost only for $T = 10$ iterations. We consider fairness parameters $\tau \in \{0.7, 0.9\}$ for our experiments. For the initial distributions Q_0 used in boosting, we fit and empirical distribution for finite domains and all DT boosting approaches – where we partition the domain into 50 bins when necessary (only in the synthetic dataset). Otherwise, we use fitted Gaussians for the neural network densities for the conditional $q_0(x \mid a) = \mathcal{N}(\hat{\mu}_a, \hat{\sigma}_a)$ in continuous domain data. To evaluate the loss function, *i.e.* train c_t, we sample twice the number of samples available from input distribution P for either either Q_{t-1} or Q_0.

Baselines For baseline comparisons, we consider the two main configurations of the max-ent algorithm presented in Celis et al. [2020]. In particular, we use statistical rate setting $\rho = 1$ and prior interpolation parameter $C = 0.5$ with a marginal vector constraint determined by: (1) the weighted mean from Celis et al. [2020, Algorithm 1] θ^w; and (2) the empirical expectation vector with the marginal set to ensure equal representation of the sensitive attribute θ^b. We denote the two distributions as M.Ent θ^c and M.Ent θ^b. We do not use this baseline in the synthetic dataset as there is no prediction task, *i.e.*, there is no statistical rate.

Results To evaluate the performance of FBDE, we need to evaluate both the fairness and approximation quality with respect to the original input distribution. Specifically, we consider (1) *representation rate* (RR), (2) *statistical rate* (SR), and (3) *KL divergence* (KL). Tables 2 reports aggregate metrics over the real-world datasets (synthetic equivalent in Appendix). The RR over boosting iterations for the COMPAS dataset (sensitive attribute=sex) for DTs are presented in Figure 5; and RR and KL over boosting iterations for the synthetic dataset are presented in Figure 4. A DT obtain from the first round of boosting in the COMPAS dataset is presented in Figure 2.

(i) **Boosting with Decision Trees.** The DT boosted densities with the exact fairness leveraging scheme ϑ_t^E maintain a high RR, with $\text{RR}(Q_T^{\text{DT}}) > 0.94$ for both real-world datasets with either fairness parameter $\tau = 0.7, 0.9$. Furthermore, the difference between KL for the different fairness parameters are minimal between these two datasets, with test KL differing by only 0.005 for COMPAS with the sex sensitive attribute. It can also be seen in Figure 5 that the RR does not change much after 5 boosting iterations on this dataset. On the other hand, in the synthetic dataset the final RR of Q_T^{DT} are very close to the τ parameter of FBDE. The DT boosted densities with relative fairness leveraging scheme ϑ_t^R
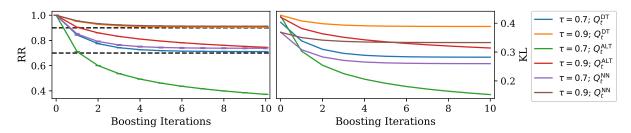
Figure 4: The KL divergence and representation rate at each iteration of boosting in the simulated mixtures of univariate Gaussian distributions. The initial representation rate of the raw data is 0.111. The error bars indicate the 0.95 confidence interval over the folds.
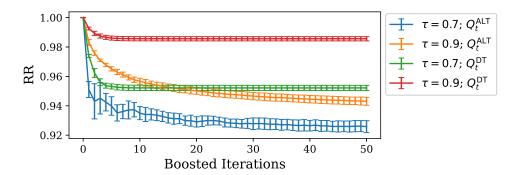

Figure 5: The representation rate of DT boosted densities in the COMPAS dataset with sex as the sensitive attribute. Error bars indicate the 0.95 confidence interval over the folds.

follow a similar story for the real-world datasets. Despite having a significantly weaker fairness guarantee in Theorem 4, $\mathrm{RR}(Q_T^{\mathrm{DT}}) > 0.92$ in these datasets. The KL is also consistently lower than the corresponding exact fairness boosting density. The weaker guarantee becomes apparent in the synthetic results, where the final RR becomes lower than the parameter τ of FBDE. As suggested by Figure 4, further decay of the RR occurs after additional boosting iterations, explicitly shown in the Appendix. When comparing to the baseline approaches from Celis et al. [2020], we often have lower fairness metrics but better KL accuracy. In particular, the RR is often slightly lower than the DT approaches – particularly when using the relative leveraging scheme. The RR is much higher across the board than our approaches, given that the max-ent method aims to ensure that this is high. However, this does not come without a cost. The KL is much higher for our approach, with the largest difference appearing in the Adult dataset; where we achieve a test KL of 0.088 versus 0.483 and 0.901 for M.Ent θ^c and M.Ent θ^b, respectively.

An additional benefit of using DT is the interpretability of c_t, in particular regarding 'collateral unfairness', additional sensitive attributes whose correlation with the target of representation rate are substantial but eventually not known. Nodes appearing in DTs ideally should not be other sensitive attributes (race when consider sensitive attribute=sex). This is especially so for the DTs' most discriminant nodes, the root, since attribute which appear can be interpreted as proxies for the original sensitive attribute Datta et al. [2017]. FBDE can be interpreted as an algorithm which finds and uses proxies to make the initial fair distribution more unfair and closer to the target (notice the drop in RR in Figure 5). In this respect Figure 2 suggests that race, also a sensitive attribute and at the root of the DT, could be factored with 'sex' in the analysis – and eventually repeating the process until the trees do not show any deemed sensitive attribute.

(ii) **Boosting with Neural Networks.** The boosted densities which utilise neural networks result in higher fairness values than the corresponding DT densities. In particular, for real-world datasets $\mathrm{RR}(Q_T^{\mathrm{NN}}) > 0.97$ over all configurations. Furthermore, when we only consider $\tau = 0.9$ then $\mathrm{RR}(Q_T^{\mathrm{NN}}) > 0.99$ for the real-world datasets. By comparing the KL with that of the initial distributions, we can see that by the end of boosting Q_T^{NN} does not change much from Q_0 compared to the DT counterparts. In the synthetic dataset, the neural network boosted densities perform more similarly to the DT boosted densities. This could be because the synthetic dataset is continuous in $\mathcal{X}$ – whereas both real-world datasets have been preprocessed to be discrete. Notably that there are two major downsides to using neural networks in FBDE: (1) the classifiers c_t are difficult to interpret; and (2) they are significantly slower to train than other methods on real-world domains (see Table 2).

6 Conclusion and Broader Impact

In this paper, we show that mollifiers of distributions, introduced in privacy, can also be used in the context of data fairness. We introduce the Fair Boosted Density Estimation (FBDE) algorithm, which can be used to debias data distributions according to the representation rate. The trade-off between fairness and boosting can be controlled by specifying the leveraging scheme of FBDE. We provide two boosting rates: one ensures that throughout boosting, produced densities will have at least a specified representation rate; whilst the other rate causes the representation rate guarantee to decay as the number of boosting iterations increase but authorises more aggressive boostiong. A key aspect of our work is that these fairness guarantees are not data dependent. We find that empirically that both leveraging scheme can be used to find distributions which maintain high representation rate, even with weaker fairness guarantee, and smaller KL divergence to target than prior work.

Ensuring fairness for ML algorithms has become a crucial problem to solve. We envision two positive outcomes of our work in addition to providing strong fairness guarantees: (1) the weak learners c_t used to construct the boosted densities can be selected for interpretability, and thus be used to put the output distributions under fairness scrutiny; and (2) the input domain of the classifiers c_t only consists of non-sensitive attributes which allows FBDE to be used in scenarios where sensitive attributes are not available at runtime or cannot be legally used for classification [Edwards and Veale, 2017]. The link we unveil between data and prediction fairness could be used to craft repositories whose use with any accuracy-compliant ML algorithms would get predictions with constraints in terms both of accuracy and fairness. There must be however care in the use of FBDE as it primarily guarantees representation rate fairness. While Lemmas 8 and 9 broaden the scope of the data fairness properties, the ultimate goal should be to design specific algorithms – perhaps based on mollifiers – to specifically accommodate for diverse fairness measures and propose as many results as necessary to link them with specific prediction fairness, as we do in Lemma 7 for equal opportunity.

References

Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems 30*, pages 3992–4001, 2017.

L. Elisa Celis, Vijay Keswani, and Nisheeth K. Vishnoi. Data preprocessing to mitigate bias: A maximum entropy based approach. In *International Conference on Machine Learning*, pages 4847–4857, 2020.

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

Hisham Husain, Borja Balle, Zac Cranko, and Richard Nock. Local differential privacy for sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 3404–3413, 2020.

Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *2012 IEEE 12th International Conference on Data Mining*, pages 924–929. IEEE, 2012.

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

Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. In *2009 IEEE International Conference on Data Mining Workshops*, pages 13–18. IEEE, 2009.

Gary King and Langche Zeng. Logistic regression in rare events data. *Political analysis*, 9(2):137–163, 2001.

James E Johndrow, Kristian Lum, et al. An algorithm for removing sensitive information: application to race-independent recidivism prediction. *The Annals of Applied Statistics*, 13(1):189–220, 2019.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

Paula Gordaliza, Eustasio Del Barrio, Gamboa Fabrice, and Jean-Michel Loubes. Obtaining fairness using optimal transport theory. In *International Conference on Machine Learning*, pages 2357–2365, 2019.

Hao Wang, Berk Ustun, and Flavio Calmon. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. In *International Conference on Machine Learning*, pages 6618–6627, 2019.

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

M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *NeurIPS'16*, pages 3315–3323, 2016.

A.-K. Menon, H. Narasimhan, S. Agarwal, and S. Chawla. On the statistical consistency of algorithms for binary classification under class imbalance. In *ICML'13*, pages 603–611, 2013.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. *And it's biased against blacks. ProPublica*, 23, 2016.

Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the compas recidivism algorithm. *ProPublica (5 2016)*, 9(1), 2016.

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, October 2018. URL https://arxiv.org/abs/1810.01943.

Dheeru Dua and E Karra Taniskidou. Uci machine learning repository [http://archive. ics. uci. edu/ml]. irvine, ca: University of california. *School of Information and Computer Science*, 2017.

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 *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1193–1210, 2017.

Lilian Edwards and Michael Veale. Slave to the algorithm: Why a right to an explanation is probably not the remedy you are looking for. *Duke L. & Tech. Rev.*, 16:18, 2017.

Appendix

Table of Contents

A Proof of main results

A.1 Proof of Lemma 1

Proof. As $\mathcal{M}$ is a ε-fair mollifier, the following constraint holds for all $Q \in \mathcal{M}$ and for all $a_i, a_j \in \mathcal{A}$ with the fair distribution $F \in \mathcal{M}$:

$$\frac{f[A = a_i]}{f[A = a_j]} \leq \exp(\varepsilon) \cdot \frac{q[A = a_i]}{q[A = a_j]} \iff 1 \leq \exp(\varepsilon) \cdot \frac{q[A = a_i]}{q[A = a_j]}$$

$$\iff \exp(-\varepsilon) \leq \frac{q[A = a_i]}{q[A = a_j]}.$$

Thus all $Q \in \mathcal{M}$ has representation rate $RR(Q) \geq \exp(-\varepsilon)$. $\square$

A.2 Proof of Lemma 2

Proof. Suppose that $Q \in \mathcal{D}(\mathcal{X} \times \mathcal{A})$ such that $RR(Q) \geq \exp(-\varepsilon/2)$ and $RR(Q_0) = 1$.

Thus for all $a_i, a_j \in \mathcal{A}$,

$$RR(Q, a_i, a_j) \geq \exp(-\varepsilon/2) \iff \frac{RR(Q, a_i, a_j)}{RR(Q_0, a_i, a_j)} \geq \exp(-\varepsilon/2)$$

$$\iff \max\left\{ \frac{RR(Q, a_i, a_j)}{RR(Q_0, a_i, a_j)}, \frac{RR(Q_0, a_i, a_j)}{RR(Q, a_i, a_j)} \right\} \geq \exp(-\varepsilon/2).$$

That is $Q \in \mathcal{M}_{\varepsilon, Q_0}$. $\square$

A.3 Efficient expectations for conditional, Equation 9

We present an equivalent trick for computing the conditional expectation $\mathbb{E}_{q_t(\cdot|a)}[g(x)]$. First note that from Eq. (6) and (10) the conditional density can be calculated as follows:

$$\begin{aligned} q_t(x \mid a) &= \frac{Q_t(x, a)}{q_t(a)} \\ &= \left(\frac{1}{Z_t} \exp(\vartheta_t c_t(x)) Q_{t-1}(x, a) \right) \bigg/ \left(q_{t-1}(a) \cdot \frac{Z_t(a)}{Z_t} \right) \\ &= \frac{\exp(\vartheta_t c_t(x))}{Z_t(a)} \cdot \frac{Q_{t-1}(x, a)}{q_t(a)} \\ &= \frac{\exp(\vartheta_t c_t(x))}{Z_t(a)} q_{t-1}(x \mid a). \end{aligned}$$

Now we can calculate the expectation:

$$\begin{aligned} \mathbb{E}_{q_t(\cdot|a)}[g(x)] &= \int_{\mathcal{X}} \prod_{k=1}^{t} \frac{\exp(\vartheta_k c_k(x))}{Z_k(a)} g(x) dq_0(\cdot \mid a) \\ &= \mathbb{E}_{q_0(\cdot|a)} \left[\prod_{k=1}^{t} \frac{\exp(\vartheta_k c_k(x))}{Z_k(a)} g(x) \right]. \end{aligned}$$

A.4 Proof of Theorem 3

Proof. Let $\vartheta_t = -\frac{1}{C 2^t} \log \tau > 0$.

Since $c_t(x) \in [-C, C]$ for all $t \in \{1, \ldots, T\}$, by taking the smallest and largest values we have that

$$\log(\tau) \cdot \left(\frac{1}{2} \right)^{k+1} < \vartheta_k c_k(x) < -\log(\tau) \cdot \left(\frac{1}{2} \right)^{k+1}$$

By taking the exponential, integrand (w.r.t. $q_{k-1}(x \mid a)$), and logarithm, we get

$$\log(\tau) \cdot \left(\frac{1}{2}\right)^{k+1} < \log \int_{\mathcal{X}} \exp(\vartheta_k c_k(x)) dq_{k-1}(x \mid a) < -\log(\tau) \cdot \left(\frac{1}{2}\right)^{k+1}$$

$$\log(\tau) \cdot \left(\frac{1}{2}\right)^{k+1} < \log Z_k(a) < -\log(\tau) \cdot \left(\frac{1}{2}\right)^{k+1}.$$

Thus by taking the largest values in the difference of the $\log Z_k(a)$

$$\log(\tau) \cdot \left(\frac{1}{2}\right)^{k} < \log Z_k(a_i) - \log Z_k(a_j) < -\log(\tau) \cdot \left(\frac{1}{2}\right)^{k}.$$

The representation rate can then be bounded below,

$$\begin{aligned}
RR(Q_T, a_i, a_j) &= \exp\left[\sum_{k=1}^{T} \log Z_k(a_i) - \log Z_k(a_j)\right] \\
&> \exp\left[\sum_{k=1}^{T} \log(\tau) \cdot \left(\frac{1}{2}\right)^{k}\right] \qquad\qquad\text{(A.1)} \\
&\geq \exp\left[\log(\tau) \cdot \sum_{k=1}^{\infty} \left(\frac{1}{2}\right)^{k}\right] \\
&= \exp\left[\log(\tau)\right] \\
&= \tau. \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\text{(A.2)}
\end{aligned}$$

Thus representation rate of Q_T is at least τ. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

A.5 Proof of Theorem 4

Proof. The proof follows identically with Theorem 3, where we obtain the following alternative to Eq. (A.1):

$$\begin{aligned}
RR(Q_T, a_i, a_j) &> \exp\left[\log(\tau) \cdot \sum_{k=1}^{T} \frac{1}{k}\right] \\
&\geq \exp\left[\log(\tau) \cdot \left(1 + \int_1^T \frac{1}{t} dt\right)\right] \\
&= \exp\left[\log(\tau) \cdot (1 + \log T)\right] \\
&= \tau^{1+\log T}.
\end{aligned}$$

Thus we have the relative representation rate bound. $\qquad\qquad\qquad\qquad\qquad\qquad\square$

A.6 Proof of Theorem 5

To use the proof of Husain et al. [2020, Theorem 5], we need to account for the difference in domain of distributions P and Q_t. First we define the distributions Q_t with respect to log-partition function $\varphi(\vartheta)$:

$$Q_t(x, a) = \frac{1}{Z_t} \exp(\vartheta_t c_t(x)) Q_{t-1}(x, a)$$
$$= \exp(\vartheta_t c_t(x) - \varphi(\vartheta)) Q_{t-1}(x, a), \tag{A.3}$$

where $\varphi(\vartheta) := \log(Z_t) = \log \int_{\mathcal{X} \times \mathcal{A}} \exp(\vartheta_t c_t(x)) Q_{t-1}(x, a) d(x, a) = \log \int_{\mathcal{X}} \exp(\vartheta_t c_t(x)) q_{t-1}(x) dx$. Note that this is not dependent on the sensitive attribute $a \in \mathcal{A}$.

Then the drop of in KL-divergence between two successive boosting iterations can be expressed as follows:

Lemma 10. The drop in KL is

$$KL(P, Q_{t-1}) - KL(P, Q_t) = \vartheta_t \cdot \mathbb{E}_p[c_t] - \log \mathbb{E}_{q_{t-1}}[\exp(\vartheta_t c_t)]. \tag{A.4}$$

Proof. The drop can be characterised as follows,

$$KL(P, Q_{t-1}) - KL(P, Q_t) = \int_{\mathcal{X} \times \mathcal{A}} \log\left(\frac{P}{Q_{t-1}}\right) dP - \int_{\mathcal{X} \times \mathcal{A}} \log\left(\frac{P}{Q_t}\right) dP$$

$$= \int_{\mathcal{X} \times \mathcal{A}} \log\left(\frac{P}{Q_{t-1}}\right) dP - \int_{\mathcal{X} \times \mathcal{A}} \log\left(\frac{P}{\exp(\vartheta_t c_t - \varphi(\vartheta)) Q_{t-1}}\right) dP$$

$$= \int_{\mathcal{X} \times \mathcal{A}} \log\left(\frac{\exp(\vartheta_t c_t(x) - \varphi(\vartheta)) Q_{t-1}(x, a)}{Q_{t-1}(x, a)}\right) P(x, a) d(x, a)$$

$$= \int_{\mathcal{X} \times \mathcal{A}} (\vartheta_t c_t(x) - \varphi(\vartheta)) P(x, a) d(x, a)$$

$$= \int_{\mathcal{X} \times \mathcal{A}} (\vartheta_t c_t(x) - \varphi(\vartheta)) P(x, a) d(x, a)$$

$$= \int_{\mathcal{X}} (\vartheta_t c_t(x) - \varphi(\vartheta)) p(x) dx$$

$$= \vartheta_t \cdot \int_{\mathcal{X}} c_t(x) p(x) dx - \varphi(\vartheta).$$

This can be expressed in terms of expectations as

$$KL(P, Q_{t-1}) - KL(P, Q_t) = \vartheta_t \cdot \mathbb{E}_p[c_t] - \log \mathbb{E}_{q_{t-1}}[\exp(\vartheta_t c_t)]. \tag{A.5}$$

□

Notably, Lemma 10 does not depend on the sensitive attribute domain $\mathcal{A}$. In-fact, given that the weak learning assumption we use is independent on distribution on $\mathcal{X}$ (that is distributions $p, q \in \mathcal{D}(\mathcal{X})$), the proof now follows from Husain et al. [2020]. First we note that for $\tau > e^{-1}$, both leveraging schemes we propose result in $\vartheta_t < 1$ for any t. Thus, we can directly use steps of proof Husain et al. [2020, Theorem 5] in the supplementary material after Equation (17) (noting that we set $c^* = C$). This proves the theorem.

A.7 Proof of Theorem 6

To prove the theorem, we split up the proof into the upper and lower bound. In particular, we prove lemmas for a general bound and then specify the exact bound for the leveraging schemes that we propose.

A.7.1 Upper Bound

To prove the theorem's upper bounds, we first present the following general lemma. For convenience, we denote $\vartheta = (\vartheta_1, \ldots, \vartheta_T)$ and $c = (c_1, \ldots, c_T)$. We also define $\varphi(\vartheta)$ as per Section A.6.

Lemma 11. For leveraging scheme ϑ_t, suppose that $\sum_{k=1}^{T} \vartheta_k < \frac{g(T)}{2C}$ for a function $g : \mathbb{N} \to \mathbb{R}$. Then $\Delta(Q_T) \le g(T)$.

Proof. We first consider the following factoring of the KL divergence:

$$
\begin{aligned}
KL(P, Q_T) &= \int_{\mathcal{X} \times \mathcal{A}} \log \frac{P}{Q_T} \, dP \\
&= \int_{\mathcal{X} \times \mathcal{A}} \log \frac{P}{Q_0 \exp(\langle \vartheta, c \rangle - \varphi(\vartheta))} \, dP \\
&= \int_{\mathcal{X} \times \mathcal{A}} \log \frac{P}{Q_0} \, dP - \int_{\mathcal{X} \times \mathcal{A}} (\langle \vartheta, c \rangle - \varphi(\vartheta)) \, dP \\
&= KL(P, Q_0) - \int_{\mathcal{X} \times \mathcal{A}} (\langle \vartheta, c \rangle - \varphi(\vartheta)) \, dP.
\end{aligned}
$$

Thus we have

$$
\Delta(Q_T) = \int_{\mathcal{X} \times \mathcal{A}} (\langle \vartheta, c \rangle - \varphi(\vartheta)) \, dP. \tag{A.6}
$$

We will consider an upper bound for $\langle \vartheta, c \rangle - \varphi(\vartheta)$. From the upper bound on the summation, we will have the following:

$$
-C \sum_{k=1}^{T} \vartheta_k \le \sum_{k=1}^{T} \vartheta_k c_k \le C \sum_{k=1}^{T} \vartheta_k
$$

$$
\implies -\frac{g(T)}{2} < \sum_{k=1}^{T} \vartheta_k c_k < \frac{g(T)}{2}.
$$

Thus we have that

$$
-\frac{g(T)}{2} < \langle \vartheta, c \rangle < \frac{g(T)}{2}. \tag{A.7}
$$

Furthermore, given that $\varphi(\vartheta) = \log \int_{\mathcal{X} \times \mathcal{A}} \exp(\langle \vartheta, c \rangle) dQ_0$ and taking the exponential, integral, and logarithm results in a monotonic transformation, we have:

$$
-\frac{g(T)}{2} < \varphi(\vartheta) < \frac{g(T)}{2}. \tag{A.8}
$$

Thus by taking the union bound of Eq. (A.7) and (A.8) for Eq.(A.6), we have that

$$
\begin{aligned}
\Delta(Q_T) &= \int_{\mathcal{X} \times \mathcal{A}} (\langle \vartheta, c \rangle - \varphi(\vartheta)) \, dP \\
&\le \int_{\mathcal{X} \times \mathcal{A}} g(T) \, dP = g(T).
\end{aligned}
$$

As required. □

We can now consider the upper bound for each of the leveraging schemes:

- (Exact) For $\vartheta_t := \vartheta_t^E$ we have that

$$
\sum_{k=1}^{T} \vartheta_k^E = -\frac{\log \tau}{2C} \sum_{k=1}^{T} \frac{1}{2^k} < -\frac{\log \tau}{2C} \sum_{k=1}^{\infty} \frac{1}{2^k} = -\frac{\log \tau}{2C}.
$$

  Thus we have $\Delta(Q_T) \le -\log \tau$ as required.

- (Relative) For $\vartheta_t := \vartheta_t^R$ we have that

$$\sum_{k=1}^{T} \vartheta_k^R = -\frac{\log \tau}{2C} \sum_{k=1}^{T} \frac{1}{k} < -\frac{\log \tau}{2C} \left(1 + \int_1^T \frac{1}{t} dt \right) = -\frac{\log \tau}{2C} \left(1 + \log T \right)$$

Thus we have $\Delta(\widetilde{Q}_T) \leq -(\log \tau)(1 + \log T)$ as required.

A.7.2 Lower Bound

Similar to the upper bound, we first prove a general lemma. First let $\alpha(\gamma) = \Gamma(\gamma)/(\gamma C)$ and note that $C = \log 2$.

Lemma 12. For leveraging scheme ϑ_t in the HBS with fixed WLA constants, suppose that $\sum_{k=1}^{T-1} \vartheta_k \geq \frac{h(T-1)}{C}$ for a function $h : \mathbb{N} \to \mathbb{R}$, for $T > 1$. Then $\Delta(Q_T) \geq h(T-1) \cdot (\gamma_P + \alpha(\gamma_Q) \cdot \gamma_Q)$.

Proof. To prove the lemma, we repeatedly apply the KL divergence drop in the HBS.

$$
\begin{aligned}
KL(P, Q_T) &\leq KL(P, Q_{T-1}) - \vartheta_{T-1} \cdot \Lambda_{T-1} \\
&\leq KL(P, Q_{T-1}) - \sum_{k=1}^{T-1} \vartheta_k \cdot \Lambda_k \\
&= KL(P, Q_{T-1}) - \sum_{k=1}^{T-1} \vartheta_k \cdot (C\gamma_P^k + \Gamma(\gamma_Q^k)) \\
&= KL(P, Q_{T-1}) - \sum_{k=1}^{T-1} \vartheta_k \cdot (C\gamma_P^k + C\gamma_Q^k \cdot \alpha(\gamma_Q^k)) \\
&= KL(P, Q_{T-1}) - \sum_{k=1}^{T-1} \vartheta_k \cdot (C\gamma_P + C\gamma_Q \cdot \alpha(\gamma_Q)) \\
&= KL(P, Q_{T-1}) - (\gamma_P + \gamma_Q \cdot \alpha(\gamma_Q)) \cdot \left[C \sum_{k=1}^{T-1} \vartheta_k \right] \\
&\leq KL(P, Q_{T-1}) - (\gamma_P + \gamma_Q \cdot \alpha(\gamma_Q)) \cdot h(T-1).
\end{aligned}
$$

As required. $\square$

We can now consider the lower bound for each of the leveraging schemes:

- (Exact) For $\vartheta_t := \vartheta_t^E$ we have that

$$\sum_{k=1}^{T-1} \vartheta_k^E = -\frac{\log \tau}{2C} \sum_{k=1}^{T-1} \frac{1}{2^k} = -\frac{\log \tau}{2C} \left(1 - \frac{1}{2^{T-1}} \right)$$

Thus we have $\Delta(Q_T) \geq -\log \tau \cdot \left\{ \frac{\gamma_P + \gamma_Q \cdot \alpha(\gamma_Q)}{2} \left(1 - \frac{1}{2^{T-1}} \right) \right\}$ as required.

- (Relative) For $\vartheta_t := \vartheta_t^R$ we have that

$$\sum_{k=1}^{T-1} \vartheta_k^R = -\frac{\log \tau}{2C} \sum_{k=1}^{T-1} \frac{1}{k} > -\frac{\log \tau}{2C} \left(\int_1^T \frac{1}{t} dt \right) = -\frac{\log \tau}{2C} \left(\log T \right)$$

Thus we have $\Delta(\widetilde{Q}_T) \geq -\log \tau \cdot \left\{ \frac{\gamma_P + \gamma_Q \cdot \alpha(\gamma_Q)}{2} \cdot \log T \right\}$ as required.

A.7.3 Together

Proof. Combining Lemmas 11 and 12 with the two specific evaluations of leveraging schemes directly proves Theorem 6.
     $\square$

A.8 Proof of Lemma 7

Proof. Denote for short $\epsilon = p[\hat{Y} = 0 | Y = 1]$ and $q_i = p[\hat{Y} = 1 | A = a_i, Y = 1]$ so that condition (16) becomes $q_i / q_j \geq \rho$. We decompose the accuracy conditioned on class 1 as:

$$
\begin{aligned}
1 - \epsilon &= p[\hat{Y} = 1 | Y = 1] \\
&= q_1 \cdot p[A = a_1] + q_2 \cdot p[A = a_2] \\
&\leq \frac{q_1 + q_2}{1 + \tau},
\end{aligned}
\tag{A.9}
$$

where the last inequality follows from fairness: $\mathrm{RR}(Q) \geq \tau$ and $p[A = a_1] + p[A = a_2] = 1$ imply we have for $i \neq j$ $(1 - p[A = a_i])/p[A = a_i] = p[A = a_j]/p[A = a_i] \geq \tau$, implying $p[A = a_i] \leq 1/(1 + \tau)$. It follows from (A.9) that for $i \neq j$,

$$
\frac{q_i}{q_j} \geq \frac{(1 - \epsilon)(1 + \tau)}{q_j} - 1.
\tag{A.10}
$$

To get the RHS $\geq \rho$, we need

$$
1 - \epsilon \geq q_j \cdot \frac{1 + \rho}{1 + \tau},
\tag{A.11}
$$

and we would get a similar inequality replacing q_j by q_i if we had computed q_j / q_i ini (A.10). Since $q_i, q_j \leq 1$, a sufficient condition is $1 - \epsilon \geq (1 + \rho)/(1 + \tau)$, which translates into

$$
p[\hat{Y} = 0 | Y = 1] = \epsilon \leq \frac{\tau - \rho}{1 + \tau},
\tag{A.12}
$$

which is the statement of the Lemma. $\square$

A.9 Proof of Lemma 8

Proof. The statistical rate ratio can be simply expressed as the product of the ratio of the marginal and the joint distributions:

$$
\frac{p[Y = y \mid A = a_i]}{p[Y = y \mid A = a_j]} = \frac{p[Y = y, A = a_i]}{p[Y = y, A = a_j]} \cdot \frac{p[A = a_j]}{p[A = a_i]}.
$$

The joint ratio is directly lower bounded by the representation rate condition. The ratio of the marginals can be bounded by considering the maximum and minimum probabilities:

$$
\begin{aligned}
\frac{p[A = a_j]}{p[A = a_i]} &= \frac{\int_{\mathcal{Y}} p[Y = y, A = a_j] \, dy}{\int_{\mathcal{Y}} p[Y = y, A = a_i] \, dy} \\
&\geq \frac{\min\limits_{y \in \mathcal{Y}} p[Y = y, A = a_j]}{\max\limits_{y \in \mathcal{Y}} p[Y = y, A = a_j]} \cdot \frac{\int_{\mathcal{Y}} 1 \, dy}{\int_{\mathcal{Y}} 1 \, dy} \\
&= \frac{\min\limits_{y \in \mathcal{Y}} p[Y = y, A = a_j]}{\max\limits_{y \in \mathcal{Y}} p[Y = y, A = a_j]} \\
&\geq \tau.
\end{aligned}
$$

Thus together, we have

$$
\frac{p[Y = y \mid A = a_i]}{p[Y = y \mid A = a_j]} \geq \tau \cdot \tau.
$$

$\square$

A.10 Proof of Lemma 9

Proof. From Lemma 8 we have

$$\frac{p[Y = y \mid A = a_i]}{p[Y = y \mid A = a_j]} \geq \tau^2$$

$$\Longleftrightarrow 1 - \frac{p[Y = y \mid A = a_i]}{p[Y = y \mid A = a_j]} \leq 1 - \tau^2$$

$$\Longleftrightarrow 1 - \frac{p[Y = y \mid A = a_i]}{p[Y = y \mid A = a_j]} \leq \frac{1 - \tau^2}{\tau^2}.$$

Furthermore, by taking reciprocals of the statistical rate,

$$\frac{p[Y = y \mid A = a_i]}{p[Y = y \mid A = a_j]} \leq \frac{1}{\tau^2}$$

$$\frac{p[Y = y \mid A = a_i]}{p[Y = y \mid A = a_j]} - 1 \leq \frac{1}{\tau^2} - 1 = \frac{1 - \tau^2}{\tau^2}.$$

Thus we have discrimination control for $J = (1 - \tau^2)/\tau^2$.

$\square$

B Experiments

We present additional and promised results in the section.

B.1 Machine specification

Experiments were run using a machine with a Intel i7-7500U CPU @ 2.70GHz with 16GB of memory. No GPU was used.

B.2 Additional dataset descriptions

We present the following additional dataset descriptions for COMPAS and Adult:

(a) **COMPAS [Angwin et al., 2016]** The preprocessed dataset contains data where each defendant has the sex, race, age, number of priors, charge degree, and recidivism within two years. These attributes are given as binary features, where counts and continuous values are discretised into bins. The resulting dataset has a domain size of 144 and with 5,278 data points. We consider sensitive attributes of sex and race.

(b) **Adult [Dua and Karra Taniskidou, 2017]** The preprocessed dataset contains data where for each person the dataset contains the race, sex, age, years of educations, and binary label whether they earn more than 50K a year. The counts and continuous values are discretised similarly to COMPAS. The preprocessed dataset has a domain size of 504 and with 48,842 data points. We consider sex as the sensitive attribute.

B.3 Additional neural network descriptions

For the neural network boosted densities 200 epochs are used in training using the `Adam` optimiser with batch size 128, learning rate 0.001, and no weight decay.

B.4 Full tables

The full table of Table 2 containing standard deviation values can be found in Figure A.1.

Furthermore the corresponding synthetic table missing from the main text can be found in Figure A.2.

Additional to the synthetic table, we present Table A.3 and Table A.4 which reports the representation rate and KL divergence over all synthetic Gaussian mixtures tested for the neural network architecture.

B.5 Additional figures

In Figure A.4, we present the representation rate and (training) KL-divergence for boosted densities using decision trees in the COMPAS dataset with sex as sensitive attribute. This is an extension of Figure 5. We can see that even after 50 boosting iterations, the relative fairness leveraging scheme still provides reasonable representation rate values $\mathrm{RR}(\widetilde{Q}_T) > 0.925$ both possible $\tau = 0.7, 0.9$. The more aggressive boosting here allows the boosted density to get better KL divergence values without severe penalty in fairness. However, this does not hold for all datasets. Figure A.2 presents a corresponding plot where the sensitive attribute is race. Interestingly, there is a spike in representation rate and KL divergence for $\widetilde{Q}_t$ with $\tau = 0.7$. However, the boosting quickly recovers soon after.
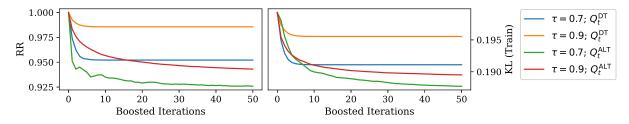

Figure A.1: Representation rate and KL divergence for the COMPAS dataset (sensitive attribute=sex) for decision tree boosted densities. Error bars not included for clarity.
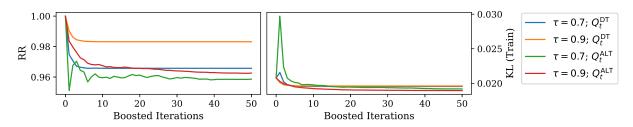
Figure A.2: Representation rate and KL divergence for the COMPAS dataset (sensitive attribute=race) for decision tree boosted densities. Error bars not included for clarity.

An additional metric we can consider is the weak learner's (WL) training accuracy over the boosting iterations. Figure A.3 presents the WL's training accuracy. We can see that the spike in performance in representation rate and KL divergence (Figure A.2) corresponds to the increase in accuracy in the WL.

Figure A.3: Weak learner accuracy for the COMPAS dataset (both sensitive attributes) for decision tree boosted densities. Error bars not included for clarity.

We further present an extend plot for Figure 4 in Figure A.4. We can see that the observation of decaying representation rate and KL divergence continues for the boosted densities using decision trees with the relative leveraging scheme. However, the boosted densities with the exact leveraging scheme maintains the fairness condition.

Figure A.4: Representation rate and KL divergence for synthetic dataset for decision tree boosted densities. Extended plot of Figure 4. Error bars not included for clarity.

Table A.1: Summary of COMPAS and Adult experiments where $T = 50$ for Q_T^{DT} and Q_T^{ALT}, and $T = 10$ for Q_T^{NN}. The mean of the measurements are report across all folds and repetitions,with standard deviation are reported in parenthesis. The representation rate of the raw data is calculated over the entire dataset.

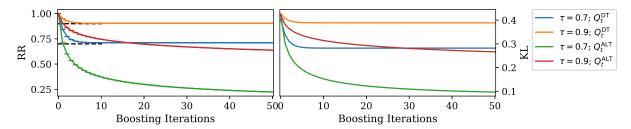|		RAW DATA	INITIAL Q_0	Q_T^{DT} (0.7)	Q_T^{DT} (0.9)	Q_T^{ALT} (0.7)	Q_T^{ALT} (0.9)	Q_T^{NN} (0.7)	Q_T^{NN} (0.9)	M.ENT θ^w	M.ENT θ^b
COMPAS SEX	RR	0.243	1.000 (0.000)	0.952 (0.002)	0.986 (0.002)	0.926 (0.005)	0.943 (0.003)	0.976 (0.001)	0.992 (0.000)	0.983 (0.009)	0.985 (0.008)
	SR	0.728	0.747 (0.123)	0.747 (0.129)	0.737 (0.132)	0.745 (0.120)	0.746 (0.125)	0.758 (0.128)	0.750 (0.124)	0.984 (0.007)	0.916 (0.014)
	KL$_{\text{(TRAIN)}}$	-	0.226 (0.003)	0.191 (0.003)	0.196 (0.003)	0.188 (0.003)	0.190 (0.003)	0.214 (0.003)	0.222 (0.003)	0.385 (0.027)	0.407 (0.025)
	KL$_{\text{(TEST)}}$	-	0.298 (0.018)	0.281 (0.028)	0.286 (0.027)	0.277 (0.029)	0.280 (0.028)	0.287 (0.018)	0.295 (0.018)	0.940 (0.164)	0.987 (0.163)
	TIME$_{\text{(MIN)}}$	-	-	4.285 (0.150)	3.439 (0.343)	3.916 (0.447)	3.634 (0.490)	19.451 (3.708)	21.561 (4.370)	0.006 (0.000)	0.007 (0.000)
COMPAS RACE	RR	0.662	1.000 (0.000)	0.966 (0.010)	0.983 (0.003)	0.959 (0.003)	0.963 (0.003)	0.988 (0.001)	0.996 (0.000)	0.977 (0.010)	0.978 (0.009)
	SR	0.747	0.536 (0.131)	0.486 (0.145)	0.494 (0.145)	0.482 (0.141)	0.483 (0.141)	0.530 (0.129)	0.534 (0.131)	0.978 (0.010)	0.852 (0.014)
	KL$_{\text{(TRAIN)}}$	-	0.048 (0.001)	0.020 (0.001)	0.020 (0.000)	0.019 (0.001)	0.019 (0.000)	0.042 (0.001)	0.046 (0.001)	0.140 (0.021)	0.139 (0.025)
	KL$_{\text{(TEST)}}$	-	0.121 (0.006)	0.110 (0.018)	0.110 (0.017)	0.110 (0.016)	0.109 (0.017)	0.115 (0.006)	0.119 (0.006)	0.464 (0.109)	0.452 (0.097)
	TIME$_{\text{(MIN)}}$	-	-	3.775 (0.213)	3.806 (0.145)	3.492 (0.391)	3.428 (0.254)	21.575 (5.063)	20.630 (4.224)	0.011 (0.009)	0.006 (0.001)
ADULT SEX	RR	0.496	1.000 (0.000)	0.949 (0.004)	0.979 (0.000)	0.939 (0.002)	0.946 (0.001)	0.987 (0.000)	0.996 (0.000)	0.987 (0.010)	0.987 (0.007)
	SR	0.360	0.382 (0.003)	0.378 (0.002)	0.367 (0.003)	0.379 (0.004)	0.377 (0.003)	0.383 (0.003)	0.382 (0.003)	0.982 (0.012)	0.873 (0.041)
	KL$_{\text{(TRAIN)}}$	-	0.061 (0.001)	0.054 (0.001)	0.055 (0.001)	0.053 (0.001)	0.053 (0.001)	0.059 (0.001)	0.060 (0.001)	0.350 (0.005)	0.696 (0.021)
	KL$_{\text{(TEST)}}$	-	0.087 (0.006)	0.089 (0.005)	0.090 (0.005)	0.088 (0.005)	0.088 (0.005)	0.085 (0.006)	0.086 (0.006)	0.483 (0.028)	0.901 (0.028)
	TIME$_{\text{(MIN)}}$	-	-	34.910 (3.536)	23.517 (0.931)	37.564 (3.523)	38.315 (1.693)	121.645 (6.339)	112.103 (6.425)	0.013 (0.010)	0.009 (0.000)

Table A.2: Summary of synthetic experiments where $T = 10$ for all boosted densities. The mean of the measurements are report across all folds and repetitions, with standard deviation are reported in parenthesis. Measurements for Q_T^{DT} and Q_T^{ALT} are with respect to 50 discretised bin for parameter $x \in X$ and the corresponding KL values for these densities are on a test set.

		RAW DATA	INITIAL Q_0	Q_T^{DT} (0.7)	Q_T^{DT} (0.9)	Q_T^{ALT} (0.7)	Q_T^{ALT} (0.9)	Q_T^{NN} (0.7)	Q_T^{NN} (0.9)
SYNTH. $s = 0.9$	RR	0.111	1.000 (0.000)	0.712 (0.002)	0.905 (0.000)	0.372 (0.001)	0.745 (0.002)	0.738 (0.005)	0.913 (0.003)
	KL	-	-	0.369 (0.000)	0.282 (0.039)	0.389 (0.012)	0.096 (0.019)	0.266 (0.006)	0.260 (0.002)	0.333 (0.001)
	TIME$_{\text{(MIN)}}$	-	-	17.232 (1.234)	18.225 (1.133)	23.519 (1.250)	18.377 (1.495)	19.441 (0.124)	19.424 (0.302)

Table A.3: Synthetic results for boosted densities using the exact leveraging scheme and neural networks. The mean of representation rate measurements over synthetic tests are report across all folds and repetitions, with standard deviation are reported in parenthesis.

μ_1	μ_2	σ_1	σ_2	s	Representation Rate			
					Raw Data	Initial Q_0	Q_T ($\tau = 0.7$)	Q_T ($\tau = 0.9$)
-1.0	0.9	1.2	0.2	0.9	0.111	1.000 (0.000)	0.766 (0.002)	0.922 (0.001)
-1.0	0.9	0.8	0.6	0.9	0.111	1.000 (0.000)	0.775 (0.003)	0.927 (0.002)
-0.9	0.9	1.2	1.0	0.9	0.111	1.000 (0.000)	0.851 (0.002)	0.954 (0.001)
-0.9	0.9	1.0	1.0	0.9	0.111	1.000 (0.000)	0.835 (0.001)	0.949 (0.002)
-0.9	0.9	1.0	1.2	0.9	0.111	1.000 (0.000)	0.847 (0.001)	0.953 (0.001)
-0.8	1.0	0.8	0.2	0.9	0.111	1.000 (0.000)	0.747 (0.002)	0.915 (0.001)
-0.8	1.0	1.0	1.6	0.9	0.111	1.000 (0.000)	0.862 (0.002)	0.958 (0.001)
-0.8	1.0	0.8	0.8	0.9	0.111	1.000 (0.000)	0.802 (0.001)	0.937 (0.001)
-0.3	0.4	1.6	1.8	0.9	0.111	1.000 (0.000)	0.979 (0.002)	0.994 (0.001)
-0.3	0.5	0.2	1.4	0.9	0.111	1.000 (0.000)	0.778 (0.004)	0.926 (0.000)
-0.3	0.5	0.8	1.0	0.9	0.111	1.000 (0.000)	0.929 (0.002)	0.980 (0.001)
-0.5	0.2	1.2	0.2	0.9	0.111	1.000 (0.000)	0.821 (0.002)	0.941 (0.001)
-0.5	0.2	1.0	1.8	0.9	0.111	1.000 (0.000)	0.939 (0.004)	0.983 (0.000)
-0.5	0.2	0.8	0.6	0.9	0.111	1.000 (0.000)	0.915 (0.002)	0.974 (0.001)
-0.5	0.2	1.4	0.6	0.9	0.111	1.000 (0.000)	0.907 (0.002)	0.970 (0.000)
-0.8	0.9	0.2	1.6	0.9	0.111	1.000 (0.000)	0.768 (0.004)	0.921 (0.004)
-0.8	0.9	0.8	0.8	0.9	0.111	1.000 (0.000)	0.810 (0.000)	0.940 (0.001)
-0.7	0.2	0.4	1.4	0.9	0.111	1.000 (0.000)	0.838 (0.005)	0.947 (0.001)
-0.7	0.2	0.2	1.2	0.9	0.111	1.000 (0.000)	0.779 (0.005)	0.927 (0.001)
-0.8	0.6	0.4	0.4	0.9	0.111	1.000 (0.000)	0.745 (0.005)	0.916 (0.002)
-0.8	0.6	0.2	1.2	0.9	0.111	1.000 (0.000)	0.771 (0.002)	0.926 (0.004)
0.0	0.3	1.6	1.6	0.9	0.111	1.000 (0.000)	0.995 (0.001)	0.999 (0.000)
0.0	0.3	1.4	0.2	0.9	0.111	1.000 (0.000)	0.823 (0.002)	0.941 (0.001)
0.0	0.3	2.0	1.4	0.9	0.111	1.000 (0.000)	0.978 (0.003)	0.992 (0.000)
0.0	0.3	1.8	1.8	0.9	0.111	1.000 (0.000)	0.996 (0.001)	0.999 (0.000)
0.0	0.3	1.0	1.8	0.9	0.111	1.000 (0.000)	0.950 (0.004)	0.986 (0.000)
-1.0	0.5	1.2	1.0	0.9	0.111	1.000 (0.000)	0.877 (0.002)	0.963 (0.001)
-1.0	0.5	1.0	1.2	0.9	0.111	1.000 (0.000)	0.873 (0.002)	0.962 (0.001)
-1.0	0.5	1.0	2.0	0.9	0.111	1.000 (0.000)	0.895 (0.003)	0.969 (0.001)
0.0	0.9	1.0	0.8	0.9	0.111	1.000 (0.000)	0.916 (0.002)	0.975 (0.001)
-0.9	0.3	0.2	1.6	0.9	0.111	1.000 (0.000)	0.765 (0.005)	0.922 (0.002)
-0.9	0.2	1.8	0.2	0.9	0.111	1.000 (0.000)	0.796 (0.002)	0.932 (0.001)
-0.9	0.2	1.6	0.6	0.9	0.111	1.000 (0.000)	0.880 (0.002)	0.962 (0.000)
-0.9	0.6	0.8	0.6	0.9	0.111	1.000 (0.000)	0.810 (0.001)	0.940 (0.001)
-0.9	0.5	2.0	2.0	0.9	0.111	1.000 (0.000)	0.952 (0.002)	0.986 (0.001)
-0.5	0.7	0.4	0.2	0.9	0.111	1.000 (0.000)	0.738 (0.005)	0.913 (0.003)
-0.5	0.7	0.4	1.0	0.9	0.111	1.000 (0.000)	0.812 (0.002)	0.940 (0.002)
-0.5	0.7	0.4	1.8	0.9	0.111	1.000 (0.000)	0.808 (0.004)	0.936 (0.001)
-0.5	0.7	0.4	1.4	0.9	0.111	1.000 (0.000)	0.824 (0.004)	0.939 (0.001)
-0.5	0.7	0.6	1.2	0.9	0.111	1.000 (0.000)	0.858 (0.002)	0.958 (0.001)
-0.3	0.3	0.4	1.6	0.9	0.111	1.000 (0.000)	0.841 (0.004)	0.947 (0.001)
-0.3	0.3	0.4	1.8	0.9	0.111	1.000 (0.000)	0.831 (0.003)	0.944 (0.002)
-0.3	0.3	0.4	0.2	0.9	0.111	1.000 (0.000)	0.819 (0.001)	0.943 (0.001)
-0.3	0.3	0.8	1.4	0.9	0.111	1.000 (0.000)	0.939 (0.003)	0.983 (0.000)
-0.3	0.3	0.2	0.4	0.9	0.111	1.000 (0.000)	0.801 (0.005)	0.936 (0.003)
0.0	0.0	0.4	1.4	0.9	0.111	1.000 (0.000)	0.868 (0.004)	0.959 (0.002)
-0.3	0.9	1.4	2.0	0.9	0.111	1.000 (0.000)	0.944 (0.003)	0.984 (0.001)
-0.3	0.9	1.2	1.4	0.9	0.111	1.000 (0.000)	0.923 (0.002)	0.978 (0.001)

Table A.4: Synthetic results for boosted densities using the exact leveraging scheme and neural networks. The mean of KL divergence measurements over synthetic tests are report across all folds and repetitions, with standard deviation are reported in parenthesis.

| | | | | | | KL Divergence | | |
μ_1	μ_2	σ_1	σ_2	s	Raw Data	Initial Q_0	Q_T ($\tau = 0.7$)	Q_T ($\tau = 0.9$)
-1.0	0.9	1.2	0.2	0.9	-	0.369 (0.000)	0.273 (0.001)	0.337 (0.001)
-1.0	0.9	0.8	0.6	0.9	-	0.369 (0.000)	0.279 (0.002)	0.339 (0.001)
-0.9	0.9	1.2	1.0	0.9	-	0.369 (0.000)	0.313 (0.001)	0.350 (0.000)
-0.9	0.9	1.0	1.0	0.9	-	0.369 (0.000)	0.306 (0.001)	0.348 (0.001)
-0.9	0.9	1.0	1.2	0.9	-	0.369 (0.000)	0.312 (0.001)	0.350 (0.000)
-0.8	1.0	0.8	0.2	0.9	-	0.369 (0.000)	0.264 (0.001)	0.334 (0.001)
-0.8	1.0	1.0	1.6	0.9	-	0.369 (0.000)	0.320 (0.001)	0.352 (0.000)
-0.8	1.0	0.8	0.8	0.9	-	0.369 (0.000)	0.292 (0.001)	0.343 (0.001)
-0.3	0.4	1.6	1.8	0.9	-	0.369 (0.000)	0.362 (0.000)	0.366 (0.000)
-0.3	0.5	0.2	1.4	0.9	-	0.369 (0.000)	0.282 (0.001)	0.339 (0.000)
-0.3	0.5	0.8	1.0	0.9	-	0.369 (0.000)	0.345 (0.001)	0.361 (0.000)
-0.5	0.2	1.2	0.2	0.9	-	0.369 (0.000)	0.298 (0.001)	0.344 (0.000)
-0.5	0.2	1.0	1.8	0.9	-	0.369 (0.000)	0.347 (0.001)	0.362 (0.000)
-0.5	0.2	0.8	0.6	0.9	-	0.369 (0.000)	0.338 (0.001)	0.358 (0.000)
-0.5	0.2	1.4	0.6	0.9	-	0.369 (0.000)	0.334 (0.001)	0.356 (0.000)
-0.8	0.9	0.2	1.6	0.9	-	0.369 (0.000)	0.280 (0.003)	0.337 (0.002)
-0.8	0.9	0.8	0.8	0.9	-	0.369 (0.000)	0.295 (0.001)	0.345 (0.000)
-0.7	0.2	0.4	1.4	0.9	-	0.369 (0.000)	0.308 (0.001)	0.348 (0.001)
-0.7	0.2	0.2	1.2	0.9	-	0.369 (0.000)	0.283 (0.002)	0.340 (0.000)
-0.8	0.6	0.4	0.4	0.9	-	0.369 (0.000)	0.264 (0.002)	0.334 (0.001)
-0.8	0.6	0.2	1.2	0.9	-	0.369 (0.000)	0.282 (0.001)	0.339 (0.002)
0.0	0.3	1.6	1.6	0.9	-	0.369 (0.000)	0.368 (0.000)	0.368 (0.000)
0.0	0.3	1.4	0.2	0.9	-	0.369 (0.000)	0.298 (0.001)	0.345 (0.000)
0.0	0.3	2.0	1.4	0.9	-	0.369 (0.000)	0.362 (0.001)	0.365 (0.000)
0.0	0.3	1.8	1.8	0.9	-	0.369 (0.000)	0.368 (0.000)	0.368 (0.000)
0.0	0.3	1.0	1.8	0.9	-	0.369 (0.000)	0.351 (0.001)	0.364 (0.000)
-1.0	0.5	1.2	1.0	0.9	-	0.369 (0.000)	0.324 (0.001)	0.354 (0.000)
-1.0	0.5	1.0	1.2	0.9	-	0.369 (0.000)	0.323 (0.001)	0.354 (0.001)
-1.0	0.5	1.0	2.0	0.9	-	0.369 (0.000)	0.332 (0.001)	0.357 (0.000)
0.0	0.9	1.0	0.8	0.9	-	0.369 (0.000)	0.339 (0.001)	0.358 (0.000)
-0.9	0.3	0.2	1.6	0.9	-	0.369 (0.000)	0.276 (0.002)	0.338 (0.001)
-0.9	0.2	1.8	0.2	0.9	-	0.369 (0.000)	0.286 (0.001)	0.341 (0.000)
-0.9	0.2	1.6	0.6	0.9	-	0.369 (0.000)	0.323 (0.001)	0.353 (0.000)
-0.9	0.6	0.8	0.6	0.9	-	0.369 (0.000)	0.294 (0.001)	0.344 (0.000)
-0.9	0.5	2.0	2.0	0.9	-	0.369 (0.000)	0.353 (0.001)	0.363 (0.000)
-0.5	0.7	0.4	0.2	0.9	-	0.369 (0.000)	0.260 (0.002)	0.333 (0.001)
-0.5	0.7	0.4	1.0	0.9	-	0.369 (0.000)	0.300 (0.002)	0.345 (0.001)
-0.5	0.7	0.4	1.8	0.9	-	0.369 (0.000)	0.296 (0.001)	0.344 (0.001)
-0.5	0.7	0.4	1.4	0.9	-	0.369 (0.000)	0.303 (0.002)	0.345 (0.000)
-0.5	0.7	0.6	1.2	0.9	-	0.369 (0.000)	0.319 (0.001)	0.352 (0.000)
-0.3	0.3	0.4	1.6	0.9	-	0.369 (0.000)	0.309 (0.001)	0.348 (0.001)
-0.3	0.3	0.4	1.8	0.9	-	0.369 (0.000)	0.304 (0.001)	0.347 (0.001)
-0.3	0.3	0.4	0.2	0.9	-	0.369 (0.000)	0.297 (0.001)	0.345 (0.001)
-0.3	0.3	0.8	1.4	0.9	-	0.369 (0.000)	0.348 (0.001)	0.362 (0.000)
-0.3	0.3	0.2	0.4	0.9	-	0.369 (0.000)	0.294 (0.002)	0.343 (0.001)
0.0	0.0	0.4	1.4	0.9	-	0.369 (0.000)	0.319 (0.001)	0.353 (0.001)
-0.3	0.9	1.4	2.0	0.9	-	0.369 (0.000)	0.350 (0.001)	0.363 (0.000)
-0.3	0.9	1.2	1.4	0.9	-	0.369 (0.000)	0.343 (0.001)	0.360 (0.000)