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 on 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 those 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 training data is tabular, it is defined by decision trees whose interpretability can provide clues on the source of (un)fairness. Experiments display the quality of the results obtained 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, one of the most upstream sources of biases leading to downstream discrimination is obviously data itself. Based on Laplace's "Principle of insufficient reason" (see for example [Jaynes, 1957, Section 2]), 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 the max-ent proxy (equivalently minimiser of the KL divergence) to this seed that meets domain constraints on aggregated statistics such as marginals. Provided its parameterisation is adequate, the seed guarantees that the final solution is fair, and the max-ent problem brings a final solution accurate to the precision of the aggregated statistics chosen. The neat part of the approach is computational: while solving the primal max-ent involves a domain of exponential size, the dual of the max-ent is far simpler and can be solved in polynomial time, strong duality guaranteeing matching losses. This computational trick allows for a more efficient method than preceding debiasing approaches which require a constraint for each element in the domain [Calmon et al., 2017].

In fact, this nice computational feature brings its accuracy drawback: for the dual to remain tractable enough, the aggregates cannot be too fine-grained and therefore the solution to the max-ent is accurate "only" with respect to the precision of the aggregates chosen. In [Celis et al., 2020], aggregates are attributes marginals: the distribution learned is therefore maximally accurate *only* if attributes are pairwise independent, which is obviously not true – otherwise there would be no fairness problem. Additional drawbacks include tight constraints on those marginals for the guarantees to hold, the user-design of those constraints to make sure the solution does not break fairness, and a data-blurred fairness guarantee – it is indeed data-dependent, *i.e.* can require tedious search for a sweet accuracy/fairness spot.

Alternative to methods of debiasing, some methods aim to instead re-label or re-weight a dataset [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

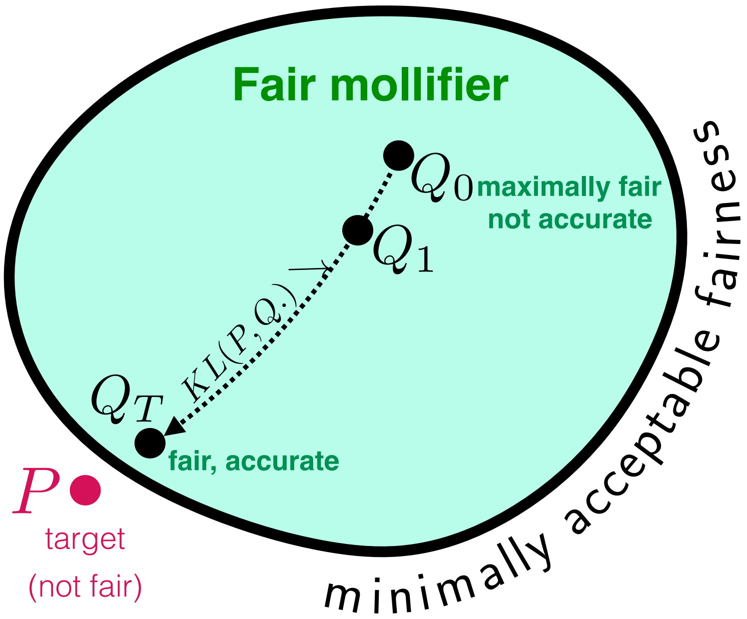

Figure 1: A mollifier $\mathcal{M}$ [Husain et al., 2020] is a set of distributions such that every pair meets a density ratio constraint. In a fair mollifier, that we introduce, this constraint is chosen in such a way that if $\mathcal{M}$ contains a perfectly fair distribution (*e.g.* uniform over 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.

attributes for trained classifiers [Johndrow et al., 2019, Feldman et al., 2015]. To achieve distribution repair, frameworks of optimal transport [Gordaliza et al., 2019] and counterfactual distributions [Wang et al., 2019] have been utilised.

Our contribution can be pretty much rooted in the same lineage of debiasing approaches but exploits a new mathematical object recently introduced in the context of differential privacy, allowing us to get an approach to fit a fair distribution[1] with better guarantees in terms of accuracy and finer (*e.g.* data-independent) guarantees in terms of fairness. This object, called a mollifier and summarised in Figure 1, is a set of distributions whose every pair meet a density ratio-constraint which, in our case, ensures fairness for every element in the set. Our second contribution involves a variation on the differentially private boosted density estimation algorithm of [Husain et al., 2020] which provides boosting-compliant convergence with respect to the KL divergence to the best fair approximator of the (non-necessarily fair) target. Our experimental contribution includes an application of the technique to tabular data for which the key elements of the Q.s are decision trees, whose interpretability provides clues on the source of (un)fairness.

The rest of this paper is organised as follows: in Section 2, we present mollifiers in the context of fairness; Section 3 presents our approach to boosting over fair mollifiers; Section 4 presents experiments, followed by two sections discussing our results (§ 5), and concluding the paper (§ 6).

2 Fair Mollifiers

Similar to the construction of locally differentially private samplers with boosting, we introduce the idea of fair mollification; which solves the problem of finding a fair distribution by projecting a distribution into a constructed set of fair distributions.

Representation Rate Fairness Suppose that we have a probability distribution $P \in \mathcal{D}(\mathcal{X} \times \mathcal{A})$ which has a domain which can be separated into sensitive $\mathcal{A}$ and non-sensitive $\mathcal{X}$ attributes. Given a $\tau \in (0, 1]$ budget, a distribution P is

[1] Similarly to and for the same reasons as [Husain et al., 2020], we conflate notations for distributions and densities with respect to some base measure.

τ-representation rate fair if for all $a_i, a_j \in \mathcal{A}$, we have

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

where $p[A = a]$ denotes the sensitive attribute marginal distribution of P.

When $\tau \to 0$, the representation rate constraint will become less restrictive, allowing for any marginal distribution. In the case when $\tau = 1$, the marginal distribution must be perfectly fair – *i.e.* uniform over the sensitive attribute(s). Representation rate fairness is the key fairness notion we study, but it can be related to several other notions [Celis et al., 2020, Williamson and Menon, 2019]. Connections with other fairness measures are developed in Section 5.

For the remainder of the paper, we define the representation rate constraint as,

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

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

Fair Mollifiers The representation rate fairness constraint only concerns itself with its own distribution. However, mollifiers introduced for locally differential privacy considers constraints over pairs of distributions. Thus, we introduce a pairwise notion of representation rate to define a fair mollifier.

Definition 1. Let $M \in \mathcal{D}(\mathcal{X} \times \mathcal{A})$ be a set of distributions and $\varepsilon > 0$. We say $\mathcal{M}$ is an **ε-fair mollifier** iff

$$\mathrm{RR}(Q, a_i, a_j) \leq \exp(\varepsilon) \cdot \mathrm{RR}(Q', a_i, a_j), \tag{3}$$

$\forall Q, Q' \in \mathcal{M}, \forall a_i, a_j \in \mathcal{A}$.

It follows that if we have a distribution $F \in \mathcal{M}$ which is perfectly fair with respect to representation rate, then all distributions in $\mathcal{M}$ will have a representation rate determined by ε.

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 $\mathrm{RR}(Q) \geq \tau$, where $\tau = \exp(-\varepsilon)$.

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 $\mathcal{M}_{\varepsilon, Q_0}$, an ε-fair mollifier, as the set of distributions Q satisfying

$$\begin{aligned} &\max\left\{\frac{\mathrm{RR}(Q, a_i, a_j)}{\mathrm{RR}(Q_0, a_i, a_j)}, \frac{\mathrm{RR}(Q_0, a_i, a_j)}{\mathrm{RR}(Q, a_i, a_j)}\right\} \\ &\leq \exp(\varepsilon/2), \end{aligned} \tag{4}$$

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

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

$$\begin{aligned} \frac{\mathrm{RR}(Q, a_i, a_j)}{\mathrm{RR}(Q', a_i, a_j)} &= \frac{\mathrm{RR}(Q, a_i, a_j)}{\mathrm{RR}(Q_0, a_i, a_j)} \frac{\mathrm{RR}(Q_0, a_i, a_j)}{\mathrm{RR}(Q', a_i, a_j)} \\ &\leq \exp(\varepsilon). \end{aligned}$$

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 $\mathrm{RR}(Q_0) = 1$. If $Q \in \mathcal{D}(\mathcal{X} \times \mathcal{A})$ has representation rate $\mathrm{RR}(Q) \geq \tau$, 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 is difficult to parameterise in closed form. However, similar to prior studies boosting methods can be used to approximate projection into the fair set of distributions.

Algorithm 1 FBDE(WL, T, τ, q_0's)

```
input: Weak learner WL, # boosting iterations $T$, representation rate $\tau$,
       initial conditional distribution $q_0(x \mid a)$
       for all $a \in \mathcal{A}$, input distribution $P$;
$Q_0(x, a) \leftarrow q_0(x \mid a) \cdot \textsc{Unif}(a)$
for $t = 1, \ldots, T$ do
    $\theta_t \leftarrow -\frac{1}{C 2^{t+1}} \log \tau$
    $c_t \leftarrow \mathrm{WL}(P, Q_{t-1})$
    $Q_t \propto Q_{t-1} \cdot \exp(\theta_t \cdot c_t)$
end for
return: $Q_T$
```

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 $\mathcal{M}$

$$\hat{P} \in \arg\min_{Q \in \mathcal{M}} \mathrm{KL}(P, Q). \tag{5}$$

In the setting of fairness, if we only want to consider distributions with representation rate at least $\tau \in (0, 1]$, we consider a relative mollifier $\mathcal{M} = \mathcal{M}_{2\varepsilon, Q_0}$ with reference distribution $\mathrm{RR}(Q_0) = 1$ and $\varepsilon = -\log \tau$.

We pick the KL divergence as it is successful in previous applications of mollifiers and its canonical use in a wide set of distributions.

3 Mollification through boosting

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.

To show the convergence of FBDE, we use similar techniques to the locally differential private's mollified boosted density estimation algorithm [Husain et al., 2020]. Concretely, we consider binary classifiers $c : \mathcal{X} \to \mathbb{R}$ which do not depend on the sensitive attribute. Furthermore, the predicted class output of these classifiers are denoted by $\mathrm{sign}(c(x)) \in \{-1, 1\}$.

In fairness, having classifiers c which do not have access to sensitive attribute has been shown to not necessarily prevent discrimination [Pedreshi et al., 2008]. Despite this, our constructed classifiers do not fall into the pitfall of relying on fairness through unawareness, as the classifier used to predict between samples between two distributions and not a target class in a dataset. Additionally, the restriction on the classifier allows the FBDE to be used in scenarios where the sensitive attribute cannot legally be used in classification [Edwards and Veale, 2017].

Similar to the boosting literature, we use the common assumption that the output of c is bounded with $c(x) \in [-C, C]$. Furthermore for technical convenience we set $c^* = \sup_{x \in \mathcal{X}} |c(x)| = C$, thereby controlling the largest value the classifier can obtain [Schapire and Singer, 1999]. We also require a notion of the central *weak learning* assumption of boosting.

Definition 2 (WLA)**.** A learner $\mathrm{WL}(\cdot, \cdot)$ satisfies the **weak learning assumption** (WLA) for $\gamma_p, \gamma_q \in (0, 1]$ iff for all $p, q \in \mathcal{D}(\mathcal{X})$, $\mathrm{WL}(p, q)$ 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 is a representation rate fairness variation of the private mollified boosted density estimation algorithm [Husain et al., 2020]. The two major deviations from the privacy algorithm are that (1) the initial distribution Q_0 needs to be a distribution with representation rate 1 and (2) the θ_t value is altered chosen to ensure fairness in the update equation.

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

$$Q_0(x, a) = q_0(x | A = a) \cdot \textsc{Unif}(a), \tag{6}$$

where $\textsc{Unif}(a)$ is the uniform distribution over sensitive attributes $\mathcal{A}$. Thus for a finite set of outcomes $\mathcal{A}$, $\textsc{Unif}(a) = \frac{1}{|\mathcal{A}|}$.

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 their sensitive attribute values. If a continuous conditional distribution is desirable, a fitted Gaussian distribution can be used.

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 θ_t

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

where Z_t is the normaliser given by

$$\begin{aligned} Z_t &= \int_{\mathcal{X}\times\mathcal{A}} \exp(\theta_t c_t(x)) Q_{t-1}(x,a) d(x,a) \\ &= \int_{\mathcal{A}} q_{t-1}(a) Z_t(a) da, \end{aligned} \tag{8}$$

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

$$Z_t(a) = \int_{\mathcal{X}} \exp(\theta_t c_t(x)) q_{t-1}(x \mid a) dx. \tag{9}$$

Using the normalisation terms expressed in Eq. 8 and Eq. 9, the sensitive attribute marginal distribution $q_t(a)$ can be expressed solely normalisation terms and the initial marginal distribution

$$\begin{aligned} q_t(a) &= \frac{q_{t-1}(a) Z_t(a)}{\int_{\mathcal{A}} q_{t-1}(a') Z_t(a') da'} = q_{t-1}(a) \frac{Z_t(a)}{Z_t} \\ &= q_0(a) \prod_{k=1}^{t} \frac{Z_k(a)}{Z_k}, \end{aligned} \tag{10}$$

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:

$$\begin{aligned} \mathrm{RR}(Q_t, a_i, a_j) &= \frac{q_t(a_i)}{q_t(a_j)} = \prod_{k=1}^{t} \frac{Z_k(a_i)}{Z_k(a_j)} \\ &= \exp\left[\sum_{k=1}^{t} \log Z_k(a_i) - \log Z_k(a_j)\right]. \end{aligned} \tag{11}$$

Using the representation rate given by Eq. 11, by specifying θ_t in FBDE, we prove that the representation rate at every iteration can be lower bounded by τ.

Theorem 3. Suppose that $\theta_t = -\frac{1}{C2^{t+1}} \log \tau > 0$. Then $\mathrm{RR}(Q_t) > \tau$.

Further, from Lemma 2 it follows that Q_t is in the relative mollifier $\mathcal{M}_{2\varepsilon, Q_0}$, for $\varepsilon = -\log(\tau)$.

We can see that fairness comes with a price as $\theta_t = 0$ when $\tau = 1$. In this case, the update equation in Eq. 18 becomes static and does not change from the initial distribution Q_0. On the other hand, when $\tau \to 0$, the update at each iteration will be large as $\theta_t \to \infty$.

Convergence guarantees Although Theorem 3 provides a guarantee on the fairness of distributions at each iteration, we have yet to show that the guarantees in convergence. It follows that by setting $C = \log 2$, we can inherit the distribution convergence rate as long as $\theta_t < 1$. This can be easily achieved by setting specifying the representation rate $\tau > e^{-1} \approx 0.368$, which in practice is high as we want to achieve fairness.

Theorem 4. 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:

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

where ($\Gamma(z) \doteq \log(4/(5-3z))$):

$$\Lambda_t = \begin{cases} c_t^* \gamma_p^t + \Gamma(\gamma_q^t) & \text{if } \gamma_q^t \in [1/3, 1] \text{ (”HBS“)} \\ \gamma_p^t + \gamma_q^t - \frac{-\log \tau}{2^{t+2}} & \text{if } \gamma_q^t \in (0, 1/3] \text{ (”LBS“)} \end{cases}.$$

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

The original theorem in [Husain et al., 2020, Theorem 5] can be directly adapted to the fairness scenario by restricting the values of C and θ_t. However, the crucial difference is how the representation rate τ parameter interacts with the update.

In the *high* boosting regime, we are guaranteed a positive drop in KL divergence. In the *low* boosting regime, we run into the same problem as the privacy case where we need $\gamma_p^t + \gamma_q^t \geq -\log \tau / 2^{t+2}$. It follows that $2^{-t} \to 0$ exponentially fast as $t \to \infty$, the low boost regime constraint vanishes after a few iterations. Notably, the rate of decay is independent to the fairness constraint – contrasting the privacy case in which it is dependent on the privacy budget. Additionally, it follows that if $\gamma_q^t > 1/4$ we will have a positive drop in KL divergence by setting $\tau = e^{-1}$ and $t = 0$ in the equation of the low boosting regime.

Additional to the iterative drop in KL divergence, we can consider a lower bound on the closeness of distributions, in a information-theoretic sense. Let $\Delta(Q) \doteq \mathrm{KL}(P, Q_0) - \mathrm{KL}(P, Q)$ and $\mathcal{M}^{\exp} \subseteq \mathcal{M}_{2\varepsilon, Q_0}$, $\varepsilon = -\log \tau$, be the subset of all possible distribution obtainable by boosting updates from Q_0 using Eq. 18. For simplicity, by fix γ_p, γ_q throughout the boosting procedure and assume that we are within HBS, we can characterise the statistical difference.

Theorem 5. If $C = \log 2$, $\tau > e^{-1}$, and FBDE is in HBS, then $\Delta(Q) \leq -\log \tau$ for $Q \in \mathcal{M}^{\exp}$ and for $T > 1$

$$\Delta(Q_T) \geq -\log \tau \cdot \left\{ \frac{\gamma_p + \gamma_q \cdot \alpha(\gamma_q)}{2} \cdot \left(1 - \frac{1}{2^{T-1}}\right) \right\},$$

where $\alpha(\gamma) = \Gamma(\gamma)/(\gamma \log 2)$.

It follows that as $\gamma_p \to 1$, $\gamma_q \to 1$, and $T \to \infty$ we will have $\Delta(Q_T) \geq -\log \tau$. Thus in combination of the first inequality, in the limit of updates, we obtain the best information-theoretic distribution from $\mathcal{M}^{\exp}$; where we are guaranteed to get progressively closer to the information-theoretic limit when $P \in \mathcal{M}^{\exp}$. Notably, when $\tau = 0$ this lower bound becomes zero. This comes from the fact that for any t we will have $\theta_t = 0$, which causing no change in distribution in each iteration. We note that the degenerate case of $\tau \to 0$ does not occur as we have the assumption of $\tau > e^{-1}$.

Weak learner with cross entropy loss One problem of using boosted density estimators is that Q_t needs to be sampled at each iteration to train the weak learner. However, if the weak learner is using the cross-entropy loss function to differentiate between the distributions, instead of sampling from Q_t we can instead sample from Q_0, which is often significantly easier. The binary cross entropy loss of classifiers c_t in the limit of data is given by

$$\mathrm{CE}(c_t) = -\mathbb{E}_P[\log(\hat{c}_t(x))] - \mathbb{E}_{Q_{t-1}}[\log(1 - \hat{c}_t(x))],$$

where $\hat{c}_t = \sigma \circ c_t$ with $\sigma = 1/(1 - e^{-1})$.

We can expand the expectation of Q_{t-1} by unrolling the recursive definition in Eq. 18:

$$\begin{aligned} &\mathbb{E}_{Q_{t-1}}[\log(1 - \hat{c}_t(x))] \\ &= \int_{\mathcal{X} \times \mathcal{A}} \prod_{k=1}^{t-1} \frac{\exp(\theta_k c_k(x))}{Z_k} \log(1 - \hat{c}_t(x)) dQ_0 \\ &= \mathbb{E}_{Q_0} \left[\prod_{k=1}^{t-1} \frac{\exp(\theta_k c_k(x))}{Z_k} \log(1 - \hat{c}_t(x)) \right]. \end{aligned} \tag{13}$$

By substituting Eq. 13 into the cross entropy loss, we only need samples from Q_0 to evaluate the loss.

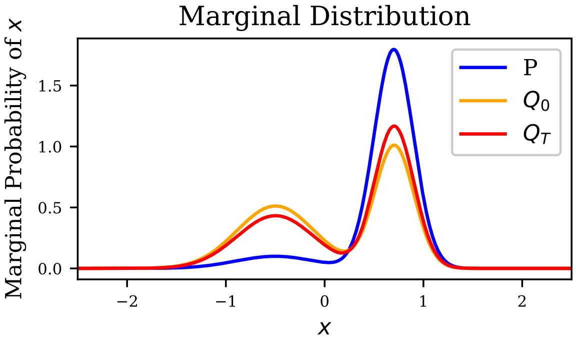

Figure 2: A comparison of the marginal distributions of initial distribution Q_0, final boosted distribution Q_T, and the simulated Gaussian mixtures P. The mixture model parameters are $\mu_1 = -0.5$, $\mu_2 = 0.7$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, and $s = 0.9$; and boosting parameters are $\tau = 0.7$ and $T = 10$.

4 Experiments

Datasets We evaluate the performance of FBDE with a simulated Gaussian mixture dataset and two standard datasets used in the fairness ML literature.

- **Simulated Gaussian Mixtures** are used to test FBDE over continuous domains. Concretely, we sample values $(x, a) \in \mathbb{R} \times \{0, 1\}$ using
$$\begin{aligned} a &\sim \textsc{Bernoulli}(s) \\ x &\sim \mathcal{N}(\mu_a, \sigma_a), \end{aligned} \tag{14}$$
where s determines the mixture balance between two normal distributions indexed by $a \in \{0, 1\}$ with mean μ_a and standard deviation σ_a. For our experiments, we consider parameter settings where our Q_0 with a perfectly fair sensitive marginal is not a good fair estimation of the original distribution. Figure 2 demonstrates one example of the miss-match between Q_0 and the true distribution that we test.
- **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], which for each defendant contains 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 separately consider the sensitive attributes of sex and race.
- **Adult [Dua and Karra Taniskidou, 2017]** The Adult dataset 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], which for each person the dataset contains the race, sex, age, years of educations, and binary label whether they earn more than $50K$ a year, where the counts and continuous values are discretised. The resulting preprocessed dataset has a domain size of 504 and contains 48,842 data points. We consider sex as the sensitive attribute.

Architectures For the weak learner of FBDE , we fit a neural network classifier for each c_t:
$$\mathcal{X} \times \mathcal{A} \xrightarrow[\text{dense}]{\text{ReLU}} \mathbb{R}^{20} \xrightarrow[\text{dense}]{\text{ReLU}} \mathbb{R}^{20} \xrightarrow[\text{dense}]{\text{sigmoid}} (0, 1), \tag{15}$$
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. 13. The number of Q_0 samples used to evaluate the loss function is twice the number of samples available from P. 200 epochs are used in training using the $Adam$ optimiser with batch size 128, learning rate 0.001, and no weight decay. For the synthetic dataset we train using 5,000 samples from P, a Gaussian mixture. We use parameter settings resulting in the large KL divergence from Q_0 in the range of $\mu_1 \in [-1, 0]$, $\mu_2 \in [0, 1]$, $\sigma_1, \sigma_2 \in [0, 2]$, and $s \in [0.7, 0.9]$.

5-fold cross validation in evaluation of the boosting algorithm, i.e. the datasets are partitioned into 5 equal parts and 4 parts are used to construct the fair distributions for each repetition. For the boosting parameters for FBDE, we pick

			Boosted Distributions			
			Raw Data	Initial Q_0	Q_T $(\tau = 0.7)$	Q_T $(\tau = 0.9)$
Synth.	$s = 0.9$	Representation Rate	0.111	1.000 (0.000)	0.738 (0.005)	0.913 (0.003)
		KL Divergence (exact)	-	0.369 (0.000)	0.260 (0.002)	0.333 (0.001)
		Runtime (min)	-	-	19.441 (0.124)	19.424 (0.302)
COMPAS	Sex	Representation Rate	0.243	1.000 (0.000)	0.981 (0.001)	0.994 (0.001)
		KL Divergence (train)	-	0.201 (0.006)	0.196 (0.006)	0.199 (0.005)
		KL Divergence (test)	-	0.282 (0.027)	0.277 (0.028)	0.285 (0.024)
		Runtime (min)	-	-	10.368 (0.070)	10.403 (0.102)
	Race	Representation Rate	0.662	1.000 (0.000)	0.988 (0.001)	0.997 (0.000)
		KL Divergence (train)	-	0.023 (0.001)	0.021 (0.001)	0.022 (0.001)
		KL Divergence (test)	-	0.105 (0.011)	0.104 (0.012)	0.108 (0.009)
		Runtime (min)	-	-	10.466 (0.092)	10.471 (0.127)
Adult	Sex	Representation Rate	0.496	1.000 (0.000)	0.987 (0.000)	0.996 (0.000)
		KL Divergence (train)	-	0.061 (0.001)	0.058 (0.001)	0.060 (0.001)
		KL Divergence (test)	-	0.087 (0.006)	0.085 (0.006)	0.086 (0.006)
		Runtime (min)	-	-	105.484 (2.165)	106.090 (2.523)

Table 1: 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. The simulated Gaussian mixture "Synth.", uses parameters $\mu_1 = -0.5$, $\mu_2 = 0.7$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, and $s = 0.9$.

$T = 10$ for the number of boosting iterations. As a result of the decay rate of θ_t, little change appears after $t > 10$ (smaller than machine precision). We consider fairness parameters $\tau \in \{0.7, 0.9\}$. Lower values are not tried as the goal is restricting the representation rate for fairness. For the initial distribution Q_0, in the synthetic Gaussian scenario we use fitted normal distributions for the conditional $q_0(x \mid a) = \mathcal{N}(\hat{\mu}_a, \hat{\sigma}_a)$. In the real world dataset scenarios, we use a fitted empirical distribution over the discrete domain.

Metrics 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 the *(1) representation rate* given by Eq. 1. to measure the fairness of boosted distributions. To test the approximation quality of boosted distributions, we use the *(2) KL divergence* between the distribution given by Eq. 14 for the synthetic dataset scenario; and the KL divergence between the empirical sampled distribution and Q_t for the real world dataset scenarios.

Results Table 1 reports the empirical results comparing FBDE across the various datasets, sensitive attribute selections, and $\tau \in \{0.7, 0.9\}$ settings; including measurements of the raw data and initial distribution Q_0. In Figure 3, the representation rate and KL divergence for the simulated Gaussian mixtures per boosting iterations is shown. For the simulated Gaussian mixtures, we only show results for parameters: $\mu_1 = -0.5$, $\mu_2 = 0.7$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, and $s = 0.9$. Additional results are shown in the Appendix.

As expected from Theorem 3, the final distributions Q_T all have representation rate above their specified τ. In general, it can be seen that as we restrict the representation rate more, the resulting boosted distribution will have a larger KL divergence. Thus, over all scenarios boosting with $\tau = 0.7$ has a smaller KL divergence than $\tau = 0.9$.

In the synthetic setting, Table 1 shows a large drop in representation rate from the boost of Q_0 to Q_T. Although the representation rate achieved does not equate the specified τ, the result is reasonably close given that the raw data only has a representation rate of 0.111. Furthermore, the boosting algorithm provides a sizeable drop in KL divergence in each specification of τ. Figure 3 demonstrate that the largest drops in representation rate and KL divergence occurs in the first iterations, when θ_t is larger.

For the COMPAS and Adult dataset, Table 1 shows minor changes from the initial distribution Q_0 after boosting. The small changes in representation rate and KL divergence are a result of the classifier c_t only performing slightly better than a coin toss (0.5 accuracy). This low accuracy is a result of the perfectly fair Q_0 being a close estimator for the initial distribution, where Q_0 has similar or smaller KL divergence to methods tested in [Celis et al., 2020]. The classifier accuracy of these cases are reported in the Appendix.

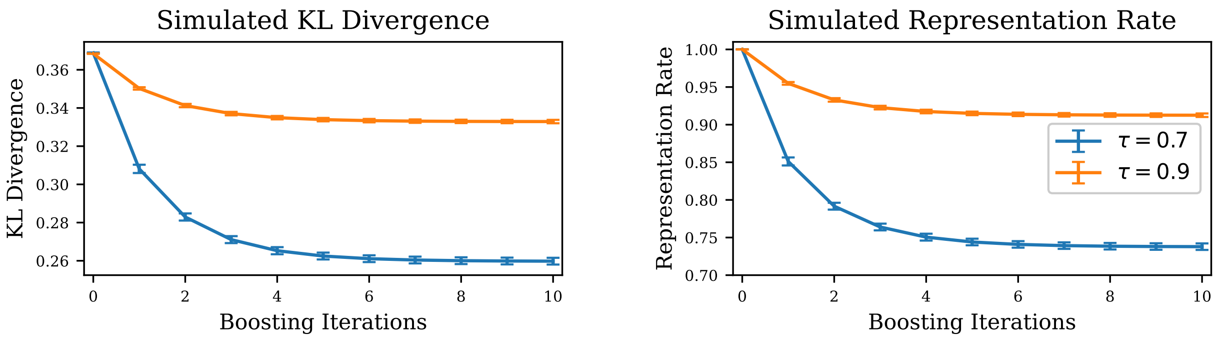

Figure 3: The KL divergence and representation rate at each iteration of boosting in the simulated mixtures of univariate Gaussian distributions. The mixture model settings in these plots are $\mu_1 = -0.5$, $\mu_2 = 0.7$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, and $s = 0.9$. The error bars indicate the 0.95 confidence interval over the folds

5 Discussion

The representation rate vs other fairness Numerous fairness measures have been studied in past work [Celis et al., 2020, Williamson and Menon, 2019] (and references within). The ones directly relevant to our work are approximate measures also data dependent – measures dependent on predictions are up to a large extent irrelevant to our work. As advocated in [Williamson and Menon, 2019], approximate fairness is preferable because otherwise it is just an ideal statement of the world and it authorises tradeoffs with accuracy.

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 attributes $\mathcal{Y}$. The statistical rate fairness constraint can be expressed by replacing the marginal distributions in Eq. 1 with conditional distribution $p[Y = y \mid A = a]$:

$$\frac{p[Y = y \mid A = a_i]}{p[Y = y \mid A = a_j]} \geq \rho, \tag{16}$$

where ρ is the statistical rate constant. The interest in considering the representation rate fairness is that it also allows to control the statistical rate fairness: if we assume that our distribution has representation rate τ for the pairs (y, a), i.e., considering (y, a) as sensitive attributes, the statistical rate can be bounded by $\rho = \tau^2$. This is formalized in the following Lemma.

Lemma 6. Suppose $P \in \mathcal{D}(\mathcal{X} \times \mathcal{Y} \times \mathcal{A})$ has representation rate τ for the pair of features $(x, a) \in \mathcal{X} \times \mathcal{A}$. Then P has statistical rate $\rho = \tau^2$.

Another data-dependent approximate measure of fairness, inspired by the "80% rule", is the *discrimination control* measure of [Calmon et al., 2017], whose base information is the same as for statistical fairness, but constraint (16) is replaced by:

$$\left| \frac{p[Y = y \mid A = a_i]}{p[Y = y \mid A = a_j]} - 1 \right| \leq J, \tag{17}$$

and we trivially get from Lemma 6 the following Lemma.

Lemma 7. Suppose $P \in \mathcal{D}(\mathcal{X} \times \mathcal{Y} \times \mathcal{A})$ has representation rate τ for the pair of features $(x, a) \in \mathcal{X} \times \mathcal{A}$. Then P has discrimination control for $J = (1 - \tau^2)/\tau^2$.

In [Calmon et al., 2017], a second measure of discrimination control is proposed, which replaces the denominator in (17) by $p[Y = y]$ and aims at making sure that the distribution of a target Y given a sensitive attribute almost matches the target's. It is not hard to check from the proof of Lemma 6 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 7. Such simple tricks would allow to show that the control of the representation rate allows to tightly control more fairness measures, where tightly means in the complete range of those fairness parameters (ρ for statistical fairness, J for discrimination control, etc.), including for example the maximal deviation between subgroups [Williamson and Menon, 2019, Section 2.3]. Importantly, the bounds are not data-dependent, unlike [Celis et al., 2020].

Weaker fairness guarantees for more aggressive boosting schemes Experiments demonstrate that the weak classifiers predictably tend to have accuracy converging to $1/2$, and convergence, for example can be quite fast, for example, Decision Trees. The proof of Theorem 3 considers boosting over an infinite horizon, but one might be able to boost for just a few steps, raising the question as how a short horizon can allow for less harsh (exponentially) decreasing updates for θ_t, yet with good guarantees on fairness. The following Lemma provides a simple answer for the updates we tested for decision trees.

Theorem 8. Suppose that $\theta_t = -\frac{2}{Ct}\log\tau > 0$. Then $\mathrm{RR}(Q_T) > \tau^{O(\log T)}$.

The proof is a simple adaptation of the proof of Theorem 3 exploiting the fact that $\sum_{t=1}^{T}(1/t) \leq 1 + \int_1^T(\mathrm{d}t/t) = O(\log T)$. In other words, should we boost for just $T = 5$ iterations, we still get a representation rate at least $\tau^{2.6}$, which can still be reasonable depending on the problem. In some cases (See *e.g* Figure 8), 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 $\tau = 0.9$.

Comparison with [Celis et al., 2020] The fact that our fairness guarantees are not data dependent unlike [Celis et al., 2020, Theorem 4.5] is a major difference between approaches. It is not the most important one. As explained in [Williamson and Menon, 2019, Section 2.3], approximate fairness allows for compromises with accuracy. 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 choosing models that are not too simple to build the sequence of sufficient statistics $c_{.}(.)$ in the exponential family which approximates P (18). Such a task can be trivial, at least during the first boosting iterations or for simulated data like ours, but it is arguably not the same game over complex real world domains and after a sufficient number of boosting rounds. This is a general downside of boosting's iterative convergence: better approximations can take much more time to get. It is a lever easier to operate than [Celis et al., 2020] because it can usually be boiled down to an acceptable complexity of models produced by the weak learner.

Connections with [Husain et al., 2020] Connections between models of fairness and differential privacy have been known for at least almost a decade [Dwork et al., 2012]. It could have been reasonable to expect a connection between the mollifiers of [Husain et al., 2020] and fairness in such a context. What is interesting in our specific case is that mollifiers are arguably *more* interesting in the context of fairness than in local differential privacy, as indeed a mollifier itself is *not* private: its *sampling* is. In the case of fairness, every element of the mollifier is already fair and can be treated as such: the solution Q_T to FBDE can be delivered as a fair distribution without the need to sample it. In the context of fairness, this difference is not anecdotical, as the knowledge of the Q_T can lead to meaningful interpretations of the sources of (un)fairness as learned by the model.

6 Conclusion

We adapt the mollifier mathematical object, initially used for privacy, for the setting of representation rate fairness. Furthermore, we provide FBDE, a boosting algorithm, which debiases the data distribution for a specified fairness constraint. Our boosting approach provides better accuracy and fairness guarantees than prior work, without being data dependent. Given that there is no de facto fairness measure, an important extension of this work includes the extension of the theory of mollification for other definitions of distribution fairness directly and mollification for multiple fairness constraints over different sensitive attributes.

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.

Edwin Thompson Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.

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.

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.

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.

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.

Robert E Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336, 1999.

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.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science*, pages 214–226, 2012.

Michael Kearns and Yishay Mansour. On the boosting ability of top–down decision tree learning algorithms. *Journal of Computer and System Sciences*, 58(1):109–128, 1999.

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 $F \in \mathcal{M}$:

$$\begin{aligned}\frac{f[A=a_i]}{f[A=a_j]} \leq \exp(\varepsilon) \cdot \frac{q[A=a_i]}{q[A=a_j]} &\iff 1 \leq \exp(\varepsilon) \cdot \frac{q[A=a_i]}{q[A=a_j]} \\ &\iff \exp(-\varepsilon) \leq \frac{q[A=a_i]}{q[A=a_j]}.\end{aligned}$$

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

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 $a_i, a_j \in \mathcal{A}$,

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

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

A.3 Proof of Theorem 3

Proof. Let $\theta_t = -\frac{1}{C2^t} \log \tau > 0$.

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

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

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

$$\begin{aligned} \log(\tau) \cdot \left(\frac{1}{2}\right)^{k+1} &< \log \int_{\mathcal{X}} \exp(\theta_k c_k(x)) dq_{k-1}(x \mid a) < -\log(\tau) \cdot \left(\frac{1}{2}\right)^{k+1} \\ \log(\tau) \cdot \left(\frac{1}{2}\right)^{k+1} &< \log Z_k(a) < -\log(\tau) \cdot \left(\frac{1}{2}\right)^{k+1}. \end{aligned}$$

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,

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

Thus representation rate of Q_T is at least τ. □

A.4 Proof of Theorem 4

We adapt the proof of [Husain et al., 2020, Theorem 5]. The primary difference which needs to be accounted for is the difference in domain of distributions P and Q_t. First we define the distributions Q_t with respect to log-partition function $\varphi(\theta)$:

$$\begin{aligned} Q_t(x, a) &= \frac{1}{Z_t} \exp(\theta_t c_t(x)) Q_{t-1}(x, a) \\ &= \exp(\theta_t c_t(x) - \varphi(\theta)) Q_{t-1}(x, a), \end{aligned} \tag{18}$$

where $\varphi(\theta) := \log(Z_t) = \log \int_{\mathcal{X} \times \mathcal{A}} \exp(\theta_t c_t(x)) Q_{t-1}(x, a) d(x, a) = \log \int_{\mathcal{X}} \exp(\theta_t c_t(x)) q_{t-1}(x) dx$.

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

Lemma 9. The drop in KL is

$$KL(P, Q_{t-1}) - KL(P, Q_t) = \theta_t \cdot \mathbb{E}_p[c_t] - \log \mathbb{E}_{q_{t-1}}[\exp(\theta_t c_t)]. \tag{19}$$

Proof. The drop can be characterised as follows,

$$\begin{aligned} KL(P, Q_{t-1}) - KL(P, Q_t) &= \int_{\mathcal{X}\times\mathcal{A}} \log\left(\frac{P}{Q_{t-1}}\right) dP - \int_{\mathcal{X}\times\mathcal{A}} \log\left(\frac{P}{Q_t}\right) dP \\ &= \int_{\mathcal{X}\times\mathcal{A}} \log\left(\frac{P}{Q_{t-1}}\right) dP - \int_{\mathcal{X}\times\mathcal{A}} \log\left(\frac{P}{\exp(\theta_t c_t - \varphi(\theta))Q_{t-1}}\right) dP \\ &= \int_{\mathcal{X}\times\mathcal{A}} \log\left(\frac{\exp(\theta_t c_t(x) - \varphi(\theta))Q_{t-1}(x,a)}{Q_{t-1}(x,a)}\right) P(x,a) d(x,a) \\ &= \int_{\mathcal{X}\times\mathcal{A}} (\theta_t c_t(x) - \varphi(\theta)) P(x,a) d(x,a) \\ &= \int_{\mathcal{X}\times\mathcal{A}} (\theta_t c_t(x) - \varphi(\theta)) P(x,a) d(x,a) \\ &= \int_{\mathcal{X}} (\theta_t c_t(x) - \varphi(\theta)) p(x) dx \\ &= \theta_t \cdot \int_{\mathcal{X}} c_t(x) p(x) dx - \varphi(\theta). \end{aligned}$$

This can be expressed in terms of expectations as

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

□

It follows that the WLA can be used to bound both terms in Lemma 9 as the WLA is with respect to classifiers $c_t : \mathcal{X} \to \mathbb{R}$ not depending on sensitive attributes. The remainder of the proof follows closely to that in [Husain et al., 2020], primarily bounding the second term $\log \mathbb{E}_{q_{t-1}}[\exp(\theta_t c_t)]$ when $C = \log 2$. We only present full proofs for particular parts of the proof of Theorem 4 which differentiates from the original proof for privacy.

First we consider when $\gamma_q^t < 1/3$ and use Hoeffding's Lemma to provide an upper bound on the drop.

Lemma 10 (Hoeffding's Lemma). Let X be a bounded random variable $a \leq X \leq b$ with distribution q such that $\mathbb{E}_q[X] = 0$, then for all $\lambda > 0$, we have

$$\mathbb{E}_q[\exp(\lambda \cdot X)] \leq \exp\left(\frac{\lambda^2(b-a)^2}{8}\right) \tag{20}$$

Lemma 11. Suppose $c^* = C = \log 2$ and $\tau > e^{-1}$. Then,

$$KL(P, Q_{t-1}) - KL(P, Q_t) > \theta_t \cdot c^* \left(\gamma_p + \gamma_q^t - \frac{-\log \tau}{2^{t+2}}\right). \tag{21}$$

Proof. By letting $X = c_t(x) - \mathbb{E}_{q_{t-1}}[c_t]$ with upper bound $b = c_t^*$ and lower bound $a = -c_t^*$, we have that

$$\mathbb{E}_{q_{t-1}}[\theta_t X] = \mathbb{E}_{q_{t-1}}[c_t - \mathbb{E}_{q_{t-1}}[c_t]] = 0. \tag{22}$$

Thus we have that

$$\exp(\theta_t X) = \exp(\theta_t c_t) \cdot \exp(\theta_t \mathbb{E}_{q_{t-1}}[-c_t]), \tag{23}$$

which provides an upper bound by Hoefffding's Lemma:

$$\begin{aligned} \mathbb{E}_{q_{t-1}}\left[\exp(\theta_t c_t) \cdot \exp(\theta_t \mathbb{E}_{q_{t-1}}[-c_t])\right] &= \mathbb{E}_{q_{t-1}}\left[\exp(\theta_t c_t)\right] \cdot \exp(\theta_t \mathbb{E}_{q_{t-1}}[-c_t]) \\ &\leq \exp\left(\theta_t^2 \cdot \frac{(c_t^*)^2}{2}\right). \end{aligned}$$

By rearranging and taking the logarithm, we have the following bound

$$\begin{aligned}\log \mathbb{E}_{q_{t-1}}\left[\exp(\theta_t c_t)\right] &\leq \theta_t^2 \cdot \frac{(c_t^*)^2}{2} - \theta_t \mathbb{E}_{q_{t-1}}[-c_t] \\ &< \theta_t^2 \cdot \frac{(c_t^*)^2}{2} - \theta_t \cdot c_t^* \cdot \gamma_q^t, \end{aligned} \tag{24}$$

where the last line follow from the WLA on q_{t-1}.

Thus drop can be bounded as follows:

$$\begin{aligned} KL(P, Q_{t-1}) - KL(P, Q_t) &= \theta_t \cdot \mathbb{E}_p[c_t] - \log \mathbb{E}_{q_{t-1}}[\exp(\theta_t c_t)] \\ &> \theta_t \cdot \gamma_p \cdot c^* - \log \mathbb{E}_{q_{t-1}}[\exp(\theta_t c_t)] \\ &> \theta_t \cdot \gamma_p \cdot c^* - \theta_t^2 \cdot \frac{(c_t^*)^2}{2} + \theta_t \cdot c_t^* \cdot \gamma_q^t \\ &= \theta_t \cdot c^* \left(\gamma_p + \gamma_q^t - \theta_t \cdot \frac{c_t^*}{2}\right) \\ &= \theta_t \cdot c^* \left(\gamma_p + \gamma_q^t + \frac{1}{c_t^* 2^{t+1}} \cdot \log \tau \cdot \frac{c_t^*}{2}\right) \\ &= \theta_t \cdot c^* \left(\gamma_p + \gamma_q^t - \frac{-\log \tau}{2^{t+2}}\right). \end{aligned}$$

□

Notably, given that $\tau > e^{-1}$ this drop will be greater than zero when $\gamma_q^t \geq 1/4$ (setting $\tau = e^{-1}$ and $t = 0$ for a lower bound). Thus we are guaranteed a larger drop as long as $\gamma_q^t \geq 1/4$, even if we are in the LBS.

When $\gamma_q^t \geq 1/3$, we use the following Lemma from [Husain et al., 2020].

Lemma 12. [Husain et al., 2020, Lemma 6 in SI] For any classifier c_t returned by Algorithm 1, we have that

$$\mathbb{E}_{q_{t-1}}[\exp(c_t)] \leq \exp(-\Gamma(\gamma_q^t)), \tag{25}$$

where $\Gamma(z) = \log(4/(5 - 3z))$.

Lemma 13. Suppose $c^* = C = \log 2$ and $\tau > e^{-1}$. Then,

$$KL(P, Q_{t-1}) - KL(P, Q_t) > \theta_t \cdot (\gamma_p \cdot c^* + \Gamma(\gamma_q^t)). \tag{26}$$

Proof. Given that $\theta_t < 1$ when $\tau > e^{-1}$,

$$\begin{aligned} KL(P, Q_{t-1}) - KL(P, Q_t) &= \theta_t \cdot \mathbb{E}_p[c_t] - \log \mathbb{E}_{q_{t-1}}[\exp(\theta_t c_t)] \\ &> \theta_t \cdot \gamma_p \cdot c^* - \log \mathbb{E}_{q_{t-1}}[\exp(\theta_t c_t)] \\ &\geq \theta_t \cdot \gamma_p \cdot c^* - \theta_t \cdot \log \mathbb{E}_{q_{t-1}}[\exp(c_t)] \\ &\geq \theta_t \cdot \gamma_p \cdot c^* + \theta_t \cdot \Gamma(\gamma_q^t) \\ &= \theta_t \cdot (\gamma_p \cdot c^* + \Gamma(\gamma_q^t)). \end{aligned}$$

□

Theorem 5 follows from Lemma 11 and Lemma 13.

A.5 Proof of Theorem 5

Proof. To show Eq. 12 we repeatedly apply the KL divergence drop in the high boosting regime. Let $\alpha(\gamma) = \Gamma(\gamma)/(\gamma \log 2)$.

$$\begin{aligned}
KL(P, Q_T) &\leq KL(P, Q_{T-1}) - \theta_{T-1} \cdot \Lambda_{T-1} \\
&\leq KL(P, Q_0) - \sum_{t=1}^{T-1} \theta_t \cdot \Lambda_t \\
&\leq KL(P, Q_0) - \sum_{t=1}^{T-1} \theta_t \cdot (c_t^* \gamma_p^t + \Gamma(\gamma_q^t)) \\
&= KL(P, Q_0) - \sum_{t=1}^{T-1} \theta_t \cdot (c_t^* \gamma_p + \Gamma(\gamma_q)) \\
&\leq KL(P, Q_0) - \sum_{t=1}^{T-1} \theta_t \cdot (c_t^* \gamma_p + \log 2 \cdot \alpha(\gamma_q) \cdot \gamma_q) \\
&\leq KL(P, Q_0) - \sum_{t=1}^{T-1} \left(-\frac{1}{C 2^{t+1}} \log \tau \right) \cdot (c_t^* \gamma_p + \log 2 \cdot \alpha(\gamma_q) \cdot \gamma_q) \\
&= KL(P, Q_0) - \frac{-\log \tau}{2} \cdot \sum_{t=1}^{T-1} \frac{1}{2^t} \cdot \frac{1}{\log 2} \cdot (c_t^* \gamma_p + \log 2 \cdot \alpha(\gamma_q) \cdot \gamma_q) \\
&= KL(P, Q_0) - \frac{-\log \tau}{2} \cdot \sum_{t=1}^{T-1} \frac{1}{2^t} \cdot \frac{1}{\log 2} \cdot (\log 2 \cdot \gamma_p + \log 2 \cdot \alpha(\gamma_q) \cdot \gamma_q) \\
&= KL(P, Q_0) - \frac{-\log \tau}{2} \cdot (\gamma_p + \alpha(\gamma_q) \cdot \gamma_q) \cdot \sum_{t=1}^{T-1} \frac{1}{2^t} \\
&= KL(P, Q_0) - \frac{-\log \tau}{2} \cdot (\gamma_p + \alpha(\gamma_q) \cdot \gamma_q) \cdot \left(1 - \frac{1}{2^{T-1}} \right),
\end{aligned}$$

where we use the fact that $\Gamma(x) = x \cdot \alpha(x) \cdot \log 2$ and the explicit geometric sum expression. □

Lemma 14. For any $T \geq 0$, let $\theta = (\theta_1, \ldots, \theta_T)$ and $c = (c_1, \ldots, c_T)$. Then

$$\log \tau \leq \langle \theta, c \rangle - \varphi(\theta) \leq -\log \tau, \tag{27}$$

where

$$\varphi(\theta) = \log \int_{\mathcal{X} \times \mathcal{A}} \exp(\langle \theta, c \rangle) \, dQ_0. \tag{28}$$

Proof. We consider the extreme values of classifiers $c_t(x) \in [-C, C]$. This gives

$$\sum_{t=1}^{T} \theta_t c_t \leq C \sum_{t=1}^{T} \theta_t \leq C \sum_{t=1}^{T} \frac{-\log \tau}{C 2^{t+1}} = \frac{-\log \tau}{2} \sum_{t=1}^{T} \frac{1}{2^t} < \frac{-\log \tau}{2},$$

and similarly,

$$\sum_{t=1}^{T} \theta_t c_t \geq -C \sum_{t=1}^{T} \theta_t \geq -C \sum_{t=1}^{T} \frac{-\log \tau}{C 2^{t+1}} = \frac{\log \tau}{2} \sum_{t=1}^{T} \frac{1}{2^t} > \frac{\log \tau}{2}.$$

Thus we have,

$$\frac{\log \tau}{2} \leq \langle \theta, c \rangle \leq \frac{-\log \tau}{2}. \tag{29}$$

By taking the exponential, integrand (w.r.t. Q_0, and logarithm of Eq. 29, we get

$$\log \int_{\mathcal{X}\times\mathcal{A}} \exp\left(\frac{\log\tau}{2}\right) \, dQ_0 \leq \log \int_{\mathcal{X}\times\mathcal{A}} \exp(\langle\theta, c\rangle) \, dQ_0 \leq \log \int_{\mathcal{X}\times\mathcal{A}} \exp\left(\frac{-\log\tau}{2}\right) \, dQ_0$$
$$\frac{\log\tau}{2} \leq \varphi(\theta) \leq \frac{-\log\tau}{2}.$$

By required inequality by taking the highest and lowest values of $\langle\theta, c\rangle$ and $\varphi(\theta)$. □

Thus by considering Lemma 14, we can characterise any Q obtained by the boosting updates. This follows from noticing that $Q_t = \exp(\langle\theta, c\rangle - \varphi(\theta))Q_0$ by unrolling the recursive definition of Q_{t-1}.

Proof. For the upper bound of the theorem,

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

□

A.6 Proof of Lemma 6

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:

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

Thus together, we have

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

□

B Additional Experiments

B.1 Neural Networks - Continuation of Primary Text

Additional experimental results are presented here. Figure 4 shows the accuracy of the classifiers used in boosting over the different iterations. Table 2 and Table 3 reports the representation rate and KL divergence over all synthetic Gaussian mixtures tested.

In Figure 4, one should notably see that the accuracy does not deviate much from 0.5, a fair coin. In other words, the initial distribution Q_0 provides a good enough guess that it is hard to differentiate samples of it from the input distribution P. Furthermore, there are even cases where the accuracy is worse than 0.5, particularly for COMPAS with race as a sensitive attributes.

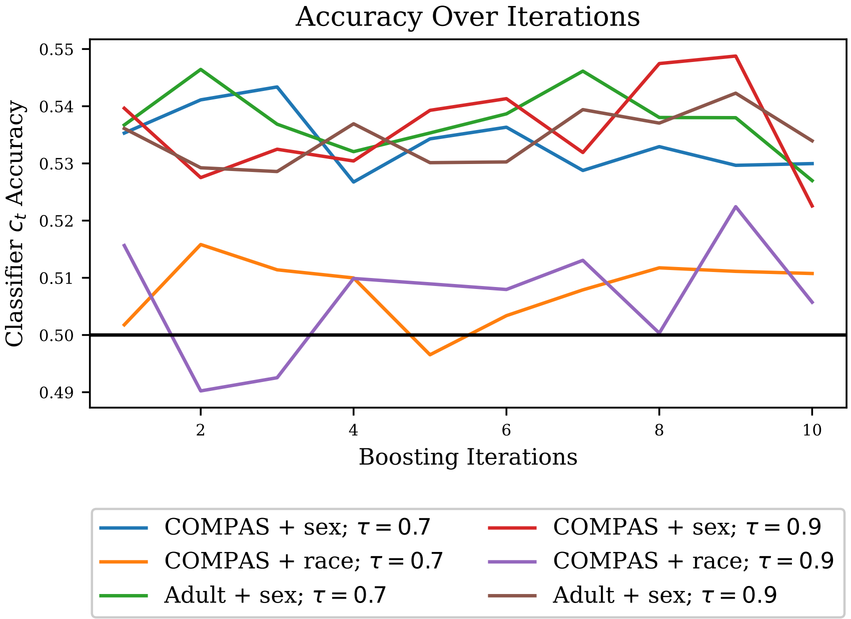

Figure 4: Accuracy over real world datasets.

					Representation Rate			
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	0.111	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 2: The mean of representation rate 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)

Table 3: The mean of KL divergence measurements over synthetic tests are report across all folds and repetitions, with standard deviation are reported in parenthesis.

B.2 Decision Trees

The following are a set of experiments utilising decision tree classifiers for the COMPAS and Adult datasets. Given that the aforementioned datasets consist of tabular data, decision trees are a natural classifier to use. We use the TopDown algorithm in [Kearns and Mansour, 1999] utilising the Gini entropy function and the set of projection functions as the node function class. Table 4 corresponds to the the decision tree results of the real world experimental settings as presented in Table 1. We use $T = 50$ boosting iterations in these decision tree experiments. The classifier accuracy over boosting iterations is also presented in Figure 5. Additionally, we present the drop in representation rate over the boosting iterations in Figure 6.

			Decision Trees Boosted Distributions			
			Raw Data	Initial Q_0	Q_T ($\tau = 0.7$)	Q_T ($\tau = 0.9$)
COMPAS	Sex	Representation Rate	0.243	1.000 (0.000)	0.953 (0.006)	0.985 (0.001)
		KL Divergence (train)	-	0.199 (0.003)	0.191 (0.003)	0.195 (0.003)
		KL Divergence (test)	-	0.290 (0.027)	0.282 (0.027)	0.286 (0.028)
		Runtime (min)	-	-	2.281 (0.156)	1.797 (0.075)
	Race	Representation Rate	0.662	1.000 (0.000)	0.968 (0.007)	0.981 (0.002)
		KL Divergence (train)	-	0.021 (0.001)	0.020 (0.000)	0.020 (0.001)
		KL Divergence (test)	-	0.111 (0.016)	0.109 (0.017)	0.110 (0.017)
		Runtime (min)	-	-	2.158 (0.026)	2.210 (0.112)
Adult	Sex	Representation Rate	0.496	1.000 (0.000)	0.949 (0.001)	0.979 (0.001)
		KL Divergence (train)	-	0.058 (0.001)	0.054 (0.001)	0.055 (0.001)
		KL Divergence (test)	-	0.093 (0.005)	0.088 (0.005)	0.090 (0.005)
		Runtime (min)	-	-	35.138 (4.410)	24.873 (3.318)

Table 4: Results of FBDE using decision tree classifiers. 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.

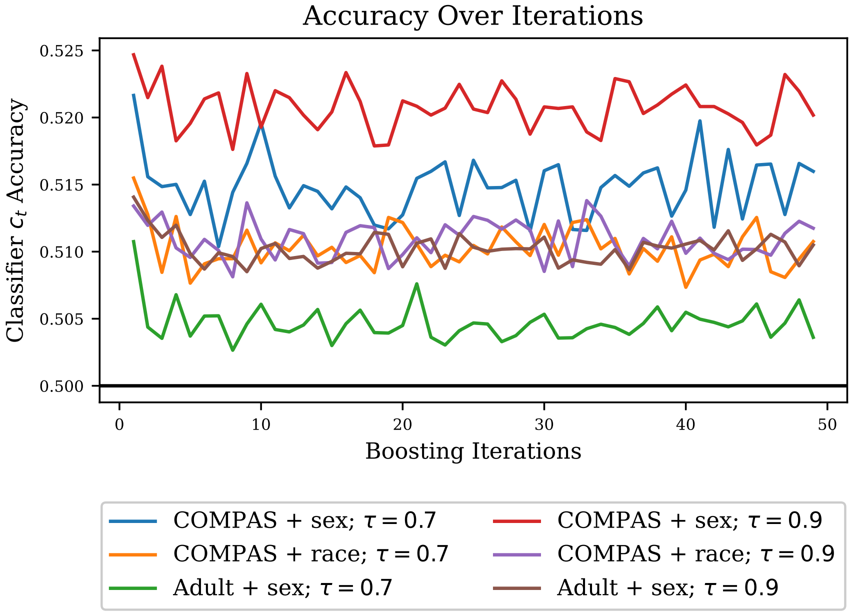

Figure 5: Decision tree classifier accuracy over real world datasets.

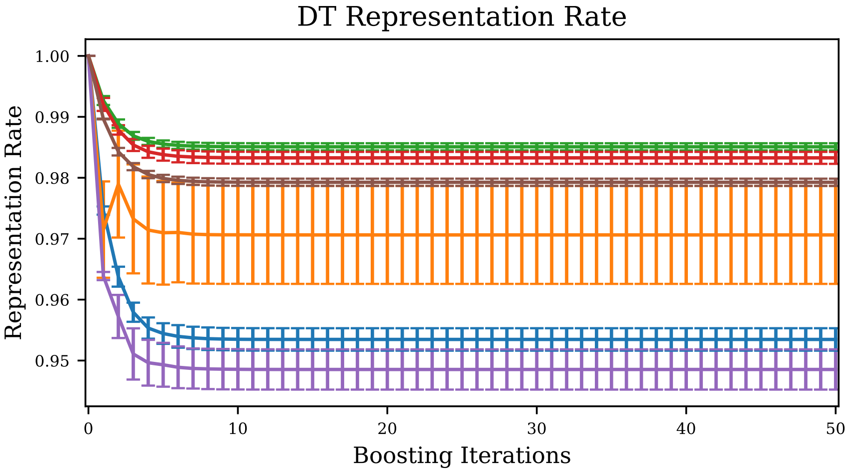

Figure 6: Decision tree representation rate over real world datasets. Colours correspond to the same settings as Figure 5

B.3 Heuristic $\tilde{\theta}_t$ with Decision Trees

The decaying weights $\theta_t = -\frac{1}{C2^{t+1}} \log \tau$, determined by a worse-case analysis, decreases in an exponential manner. As a result the changes Q_t experiences rapidly diminish after a few rounds of initial boosting. As such, we consider a "heuristic" approach where we consider $\tilde{\theta}_t := -\frac{1}{2Ct} \log \tau$. We repeat the experiments in the prior section: summarised in Table 5; accuracy over time in Figure 7; and representation rate over time in Figure 8. (We omit the legend, where the colours are consistent with Figure 5).

			Decision Trees $\tilde{\theta}$ Boosted Distributions			
			Raw Data	Initial Q_0	Q_T $(\tau = 0.7)$	Q_T $(\tau = 0.9)$
COMPAS	Sex	Representation Rate	0.243	1.000 (0.000)	0.925 (0.005)	0.946 (0.003)
		KL Divergence (train)	-	0.199 (0.003)	0.188 (0.003)	0.190 (0.003)
		KL Divergence (test)	-	0.290 (0.027)	0.280 (0.030)	0.281 (0.028)
		Runtime (min)	-	-	3.465 (0.118)	3.321 (0.211)
	Race	Representation Rate	0.662	1.000 (0.000)	0.956 (0.003)	0.962 (0.002)
		KL Divergence (train)	-	0.021 (0.001)	0.019 (0.001)	0.019 (0.000)
		KL Divergence (test)	-	0.111 (0.016)	0.110 (0.018)	0.109 (0.017)
		Runtime (min)	-	-	3.394 (0.161)	3.258 (0.095)
Adult	Sex	Representation Rate	0.496	1.000 (0.000)	0.940 (0.001)	0.943 (0.001)
		KL Divergence (train)	-	0.058 (0.001)	0.053 (0.001)	0.053 (0.001)
		KL Divergence (test)	-	0.093 (0.005)	0.088 (0.005)	0.088 (0.005)
		Runtime (min)	-	-	41.387 (3.606)	47.458 (3.919)

Table 5: Results of FBDE using decision tree classifiers and heuristic $\tilde{\theta}_t = -\frac{1}{2Ct} \log \tau$. 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.

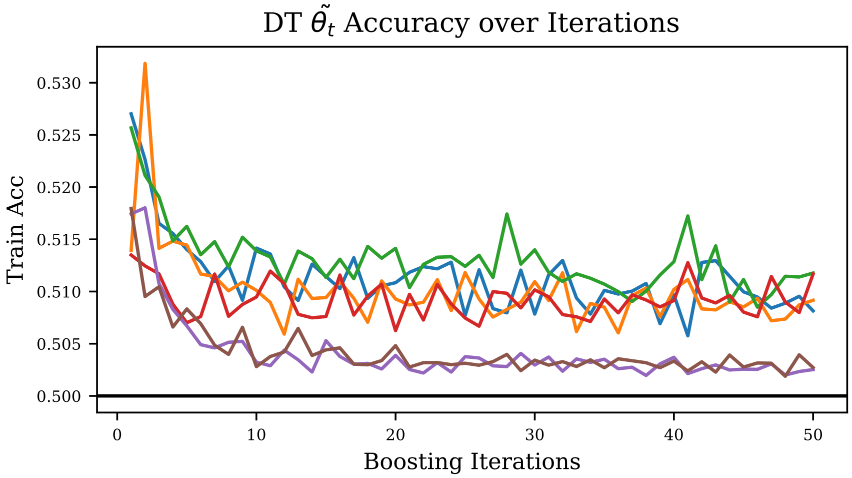

Figure 7: Decision tree classifier accuracy over real world datasets as a function of the boosting iteration. Remark that the trees accuracy tends to decrease with the iteration, showing it is harder to tell apart P from $Q_{\cdot}$. Colours correspond to the same settings as Figure 5

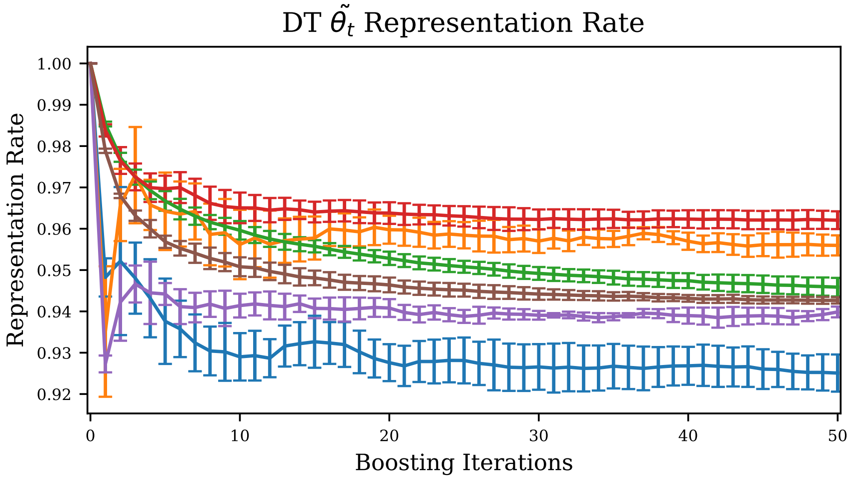

Figure 8: Decision tree representation rate over real world datasets. Colours correspond to the same settings as Figure 5

B.4 Decision Tree Analysis

Given the decision trees presented in Section B.2 and Section B.3, we examine the attributes which the decision trees chooses to split at its root nodes. In particular, these attributes can be considered as important to recover the sensitive attribute, which has been normalised to be uniform across the distribution.

For the decision tree with the proposed decay weight θ_t, Figure 9 and Figure 10 presents the histograms of the attribute choices the decision tree splits at over for $T \leq 10$ boosting iterations. We restrict the number of boosting iterations to examine the decision trees which have the largest contribution in the boosted distribution. For COMPAS, Figure 9 shows that with sex as the sensitive attribute, the primary attributes for $\tau = 0.9$ to recover the attributes are: a high number of prior counts `prior_counts>3` and `two_year_recid` to a lesser case. It is also apparent, that with lower τ values the variety of attribute splits is increased. Having a larger variety of splitting attributes for lower τ values is also consistent for when the sensitive attribute is race. When $\tau = 0.9$, the primary attributes used to recover the sensitive attribute race are: `prior_counts>3` and `age_cat=>45`. For $\tau = 0.7$, the number of decision tree splits for `age_cat=>45` increases along with the other attributes. For the Adult dataset in Figure 10, `race` and having a high age `Age<70` are the primary attributes for recovering the sensitive attribute of sex. Notably, the aforementioned attributes are solely counted as the first split in the decision trees for $\tau = 0.9$.

When we use the heuristic weight decay $\tilde{\theta}_t$, the primary attributes used to recover sensitive attributes are consistent when using weight decay θ_t. We consider the root decision tree splits up to $T \leq 10$ boosting iterations, for the same reasons for the original θ_t. For COMPAS, Figure 11 has similar splitting attributes for the different settings. However, the variety of different attributes used to as the root node in the decision trees is increased when compared to Figure 9. Noticeably, with the race sensitive attribute, `age_cat=>45` and `c_charge_degree=F` has a larger number of attribute splits for both $\tau = 0.7$ and $\tau = 0.9$. For Adult, Figure 12 has a higher variety of splitting attributes than the original decision trees. In particular, the `Age>=70` attribute is selected more than the original histogram.

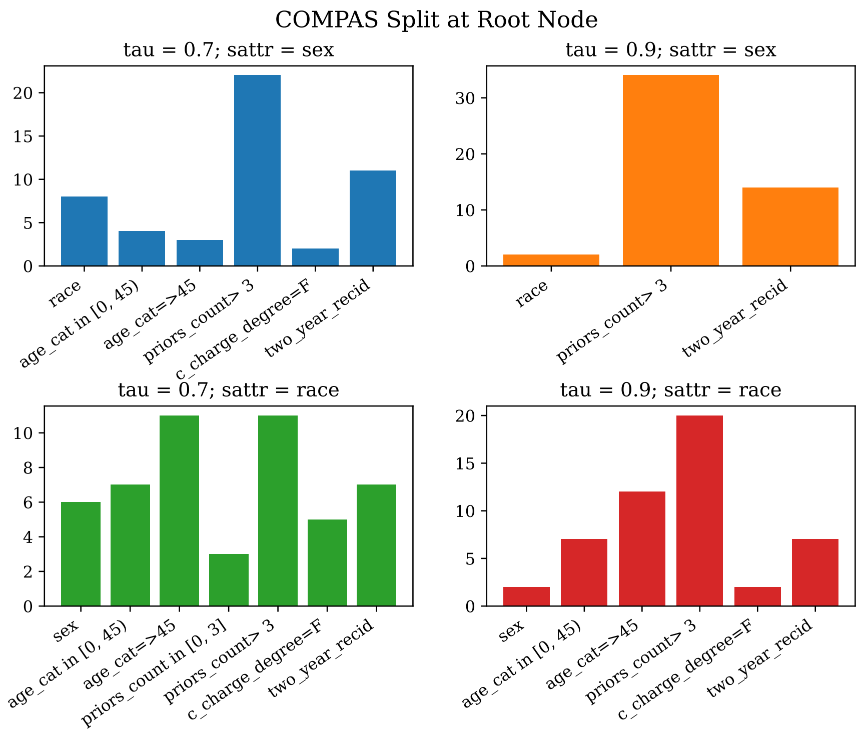

Figure 9: The decision tree root attribute splitting for the COMPAS dataset using θ_t.

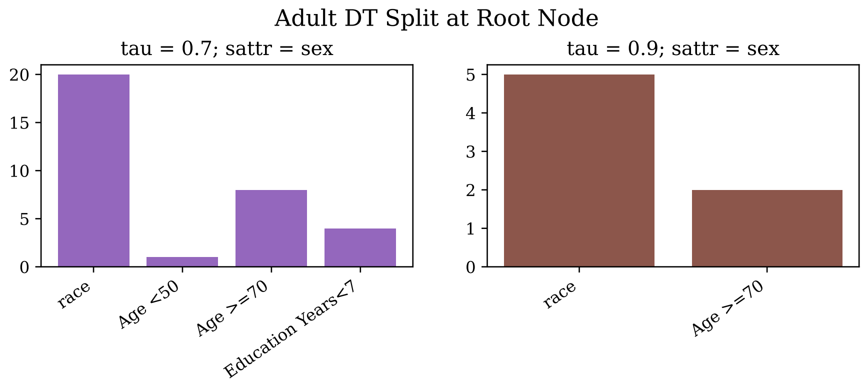

Figure 10: The decision tree root attribute splitting for the Adult dataset using θ_t.

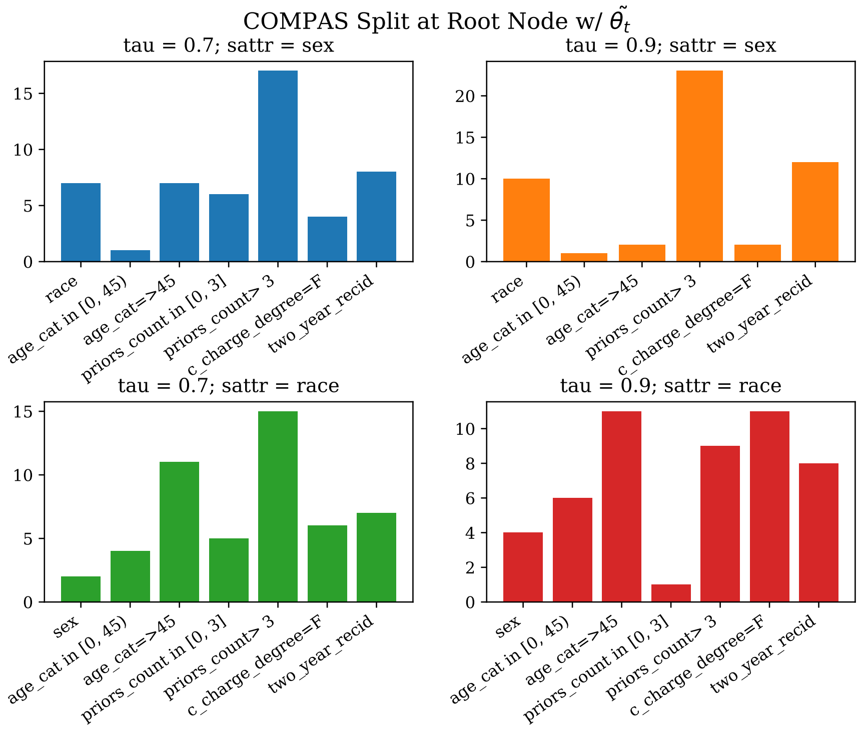

Figure 11: The decision tree root attribute splitting for the COMPAS dataset using $\tilde{\theta}_t$.

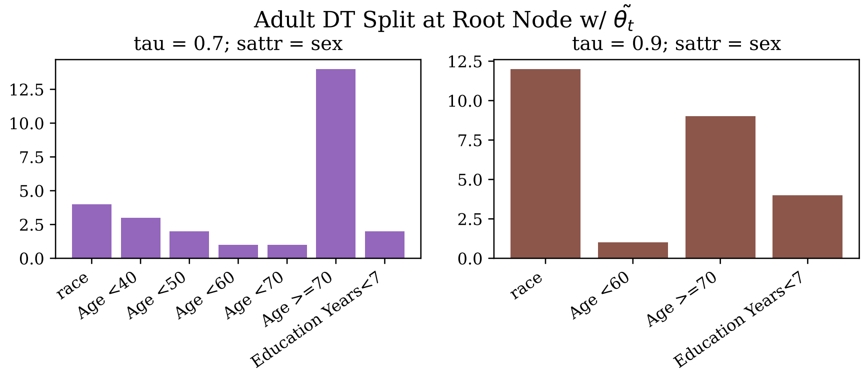

Figure 12: The decision tree root attribute splitting for the Adult dataset using $\tilde{\theta}_t$.