Auditing for Multi-Differential Fairness of Black Box Classifiers

February 2019

1 Introduction

Machine learning algorithms are more and more used to support decisions that have adverse consequences on an individual's life: for example, classifiers have supported the judicial system to decide whether a criminal offender is likely to recommit a crime; or lender to determine the default risk of a potential borrower. At issue is whether classifiers are fair in the sense of [6] , that is whether classifiers' outcomes are independent of *exogenous irrelevant characteristics* ([6]) or protected attributes, like race or gender. Abundant examples of classifiers' discrimination can be found in many applications (see [5], [2], [3]). A ProPublica story ([3]) reported that a machine learning based risk assessment tool, widely used in the United States, COMPAS, assigns higher risk to Afro-American defendants; and lower risk to Caucasian defendants.

Contestability is challenging for potential victims of machine learning discrimination because (i) many assessment tools are proprietary and are not always required to be transparent about their functioning; and, (ii) there is, at least in the United States, a paper trail of legal cases that have put the burden on the plaintiff to demonstrate disparate treatment, that is to establish that characteristics irrelevant to the task affect the algorithm's outcomes (e.g. see *Ricci et al. vs DeStefano et al.*[1], *Loomis vs. the State of Wisconsin* [4]). In *Loomis vs. the State of Wisconsin* ([4]) , the Wisconsin Supreme Court rules in favor of the use of COMPAS in recidivism risk assessment because, among other issues with the case, the court found that the plaintiff "failed to meet his burden of showing that the sentencing court actually relied on gender as a factor in sentencing".

Although a growing literature has introduced individual-level (see [12]) or small group-level (see [18], [20], [22]) notion of algorithmic fairness, there is no theoretical framework nor practical tool to characterize the individuals who can make a strong claim for being discriminated. Putting a face on a classifier's discrimination is necessary not only to empower the victims of such discrimination, but also to support the decision-making process by providing a classifier's user warnings for individual instances for which significant profiling has been detected.

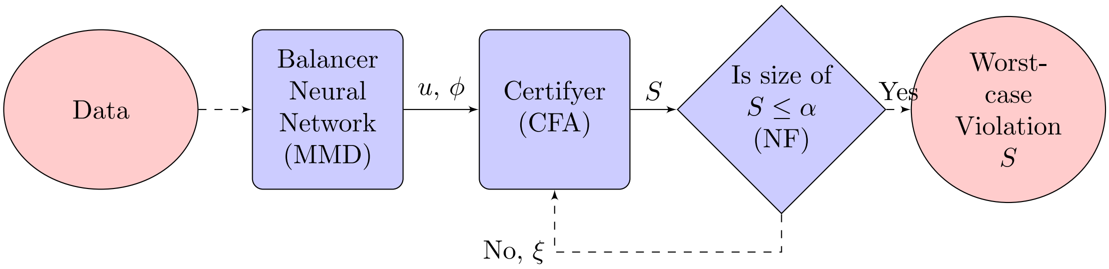
At issue is that because of social or historical reasons, data with social-related issues show imbalanced feature distributions conditional on being in a majority or minority group. For example, in the United States, individuals self-identified as Afro-Americans are more likely to have a criminal history. The objective of this paper is to find instances for which the classifier under audit stereotypes the most the imbalance observed in the data. We introduce a framework, differential fairness, to formalize the distinction between classifier's discrimination and discrimination already encoded in the data. Differential fairness guarantees that a classifier's outcomes are nearly independent of protected attributes conditional on features relevant to the decision making process. The term "differential" is used to emphasize that a classifier is fair if its outcomes are statistically nearly identical for two individuals that differ only by their protected attribute. The concept borrows from the differential privacy literature (see [13]). From a privacy perspective, differential fairness measures the incremental amount of information related to protected attributes that is leaked by the classifier and that was not leaked by the data itself.

To identify worst-case violations of differential fairness, we first relax the definition of differential fairness to a notion of multi-differential fairness that guarantees that for any sub-populations the distance between the prior and posterior distributions of protected attributes are nearly identical when measured with a maximum divergence distance. Secondly, we reduce the problem of finding worst-case violations to both a matching distribution and agnostic problem as shown in Figure 1. MMD is a neural network that matches the distributions of features conditional on protected attributes. Its objective function is the maximum-mean discrepancy between distributions conditioned on different protected attributes. In detail, the maximum-mean discrepancy measures the distance between two distributions by their kernel in a Reproducing Kernel Hilbert Space. Borrowed from the literature on covariate shifts and domain adaptation, the application of the concept of kernel-based matching in algorithmic fairness is novel and shows superior performances in terms of balancing conditional distribution than importance sampling-based methods.

Once MMD has matched conditional distributions , a second component, CFA, detects violations of differential fairness. CFA is based on a theoretical reduction of the problem of certifying the lack of differential fairness into a agnostic learning problem. Intuitively, assume that both the protected attributes and the classifier's outcomes are binary variables equal to 1 or -1. A violation of differential fairness is a group of individuals for which the classifier's outcomes is more often equal to 1 for individuals with a protected attribute equal to 1 and equal to -1 for individuals with a protected attribute equal to -1. Therefore, violations of differential fairness can be identified as subsets of the features space where protected attributes and classifier's outcomes "agrees" (i.e., their product equals $+1$). This observation allows to reduce searching for instances of differential unfairness into predicting where classifier's outcomes and protected attributes are equal. In fact, CFA uses standard classification techniques, including support vector machine or random forest tree. Moreover, our reduction allows us to derive small sample properties of our auditing approach.

CFA detects all violations of differential fairness but does not extract the most-severe ones. Theoretically, we show that finding the worst-case violations implies PAC learning a class of indicators that encodes our representation of sub-population. This is challenging since we do not directly observe the values of the function we attempt to predict. Our approach NF incrementally infuses noise to force the output from the previous block CFA to be equal to -1 wherever there is no or little violation of differential fairness. In those cases, there are as much agreements between protected attributes and classifier's outcomes as disagreements, which makes it difficult for a learner to make a prediction. By gradually increasing the weight of samples where protected attribute and classifier's outcomes disagrees, we introduces noise in the data that removes the least unfair instances so that CFA only detects the wort-case violations. We establish that our algorithm correctly identifies the sub-population victim if the most severe violation of differential fairness.

Put together, $MMD-CFA-NF$ formss a tool [**name**] that efficiently identifies the individuals who are the most discriminated by a classifier. [**name**] can work for any black box classifier whose design, inputs and parameters are unknown to the auditor. For example, when applied to the COMPAS assessment tool, [**name**] finds that individuals who have the strongest claim for discrimination are Afro-Americans with little criminal and misdemeanor history: their recidivism risk is three times higher than for individuals who are similar, but are not Afro-American.



S: sub-population, u: weights, α: minimum size, ξ: noise infusion parameter

Figure 1: Architecture

Related Work Among the rapidly growing number of contributions on algorithmic fairness (see [8] for a survey), we find three axis of work that are the most related to our paper. First, our paper, as in [18], [22], or [20], provides a definition of fairness that protects group of individuals as small as computationally possible. Empirical observations in [20] or [12]) support defining fairness at the individual level since aggregate level fairness cannot protect specific sub-populations against severe discrimination. We provide evidence in our experiments on real datasets that aggregate-level notions of fairness offer no guarantee for smaller sub-populations to be severely discriminated against. Our paper follows [18], [22], or [20] and observes that our definition of fairness, differential fairness, cannot be efficiently audited for at the individual level. In fact, as in [18], [22], by reducing the problem of certifying the lack of differential fairness to an agnostic learning problem, we show that some structure on the sub-populations we audit for are necessary to be able to identify violations of differential fairness with a polynomial number of samples.

Secondly, our paper relates to a stream of literature that interprets algorithmic unfairness as a leakage of information related to protected attributes. Our setup is mostly inspired by the literature on differential privacy (see [13]) that guarantees that a database query does not leak additional information on a given individual. Here, differential fairness is a guarantee that a classifier's outcome does not leak information that was not yet contained in the features. Few other contributions have borrowed from the differential privacy literature (see [19] and [**?**]), but none of those contributions consider the problem to detect the most severe violations. Moreover, [14], [11] define disparate impact by how much a protected attributes can be predicted. Here, we are interested by how much more predictable protected attributes are once a classifier's outcomes have been observed. We see this additional leakage as a measure of how a classifier exacerbates the discrimination already encoded in the data. Therefore, our focus is on disparate treatment over disparate impacts. Given legal precedents, at least in the United States (see [4], [1]), it seems that tools that detect potential algorithm's disparate treatments offer more opportunity for recourse than those certifying disparate impact. Moreover, disparate impact as in [14] is mitigated by rebalancing the distribution of features conditional on protected attributes. Our kernel matching technique, MMD, borrowed from the literature on covariate shift (see for example [17], [10]), on domain adaptation (see) or on counterfactual analysis (see) is an alternative technique to the transportation approaches proposed by [14] or [11].

To the extent of our knowledge, there is no existing work on characterizing the most severe instances of algorithmic discrimination. However, there is a growing number of contributions that offer tools to bolster individual recourses. For example, [**?**] and [**?**] develop algorithms to answer what-if questions. However, unlike with **name**, when transferring knowledge across protected attributes, they do not account for the distribution imbalance of the conditional distributions of features.

Contributions Our contributions are as follows:

- We introduce a concept of differential fairness to measure how a classifier leaks information related an individual's protected attribute.
- We develop a tool, **name**, that characterizes the individuals who are the most severely discriminated by a black-box classifier.
- We show that **name**'s properties are grounded into two theoretical results. Our technique to match conditional distributions of features minimizes a generalization upper bound. Moreover, we reduce the problem of certifying for the lack of differential fairness to the problem of predicting when a binary attribute contribute coincides with the classifier's outcome. This reduction allows to efficiently assess whether the exists a sub-population for which the classifier leaks information related to the distribution of its protected attribute. We show correctness and sample complexity for our certifying algorithm.
- We test the validity of our algorithms, certifying and worst violation, with synthetic datasets.
- We apply **name** to a case study a recidivism risk assessment in Broward County, Florida and identify that Afro-American with little criminal history were the most discriminated by the risk assessment. We apply **name** to four other datasets and find that

2 Individual and Multi-Differential Fairness

2.1 Preliminary

Notations An individual i is defined by a tuple $((x_i, a_i), y_i)$, where $x_i \in \mathfrak{X}$ denotes individual i's audited features; $a_i \in \mathfrak{A}$ denotes her protected attribute; and $y_i \in \{-1, 1\}$ is the classification provided by a black box classifier f. The auditor draw samples $\{((x_i, a_i), y_i)\}_{i=1}^{m}$ of size m from a distribution D on $\mathfrak{X} \times \mathfrak{A} \times \{-1, 1\}$.

Features in $\mathfrak{X}$ are not necessarily the ones used to train f. First, the auditor may not have access to all features used to train f. Secondly, the auditor may decide to leave deliberately some features used to train f out of $\mathfrak{X}$ because she believes that those features should not be used to define similarity among individuals. For example, if f classifies loan according to their probability of repayment, the auditor may consider that credit score should be used to define individual similarity, but that zipcode, because correlated with races, should not be an auditing feature, although it was used to train f.

Assumptions In our analysis we make the following assumption:

Assumption 1. *For all $x \in \mathfrak{X}$, $Pr[A|X = x] > 0$.*

Assumption 1 guarantees that the distribution of auditing features conditional on protected attributes have common support: there is no $x \in \mathfrak{X}$ that reveals perfectly the individual's protected attribute.

2.2 Individual Differential Fairness

Individual Differential Fairness We define differential fairness as the guarantee that conditional on features relevant to the tasks, a classifier's outcome is nearly independent of protected attributes:

Definition 2.1. *(Individual Differential Fairness) For $\delta \in [0, 1)$, a classifier f is $\delta-$ differential fair if for all $x \in \mathfrak{X}$ and all $a \in \mathfrak{A}$ and for all $y \in \{-1, 1\}$*

$$e^{-\delta} \leq \frac{Pr[Y = y|A = a, x]}{Pr[Y = y|A \neq a, x]} \leq e^{\delta} \tag{1}$$

The parameter δ controls how much the distribution of the classifier's outcome Y depends on protected attributes A conditional on auditing features x: larger value of δ implies larger leakage.

Differential Fairness and Distance of Distributions Since the space of auditing features $\mathfrak{X}$ does not necessarily correspond to the one used to trained the classifier f, for any $x \in \mathfrak{X}$, the auditor access samples drawn from two distributions $Y|A = a, x$ and $Y|a \neq a, x$. Individual fairness constraints these two distributions to be nearly identical when distribution similarity is defined by max-divergence. Formally, the max divergence of two distributions P and Q is defined as

$$D_{\infty}(P||Q) = \max_{y \in Y} \ln \left(\frac{Pr[P = y]}{Pr[Q = y]} \right) \tag{2}$$

The fairness condition in 2.1 can be equivalently rewritten in terms of max divergence as:

$$D_{\infty}((A|x, Y)||(A|x)) \leq \delta \tag{3}$$

If a classifier is δ-individual fair, the posterior and prior distribution of protected attribute conditional on auditing features is bounded by δ.

Relation with Differential Privacy There is an analogy between individual differential fairness for classifiers and differential privacy for database queries. Differential privacy (see [13])guarantees that outcomes from a query are not distinguishable when computed on two adjacent databases that differs only by one record. The fairness condition (1) implies that outcomes from a classifier are not distinguishable for individuals that differ only by their protected attributes. The max-divergence equivalence in Eq. (3) shows that differential fairness bounds the information leakage caused by Y conditional on what is already leaked by the auditing features x. Our application of differential privacy differs from [19] since we are looking at an individual-level definition and relies on max-divergence as a measure of distance between distributions. Our definition also differs from [**?**], since our notion of differential fairness bounds the information leaked by the classifier's outcomes conditional on the information already leaked by the auditing feature x. Therefore, our use of a differential privacy framework allows to audit whether a classifier exacerbates the unfairness already encoded in the data.

Individual Fairness The definition Eq. (2.1) is an individual level definition of fairness, since it conditions the information leakage on auditing features x. Compared to the notion of individual fairness in [12], individual differential fairness does not require to explicit a similarity metric. This is important because defining a similarity metric has been the main limitation of applying the concept of individual fairness (see a discussion on the challenges of individual fairness in [8]). The similarity of treatment in differential fairness is defined in a statistical sense as the max-divergence distance between the distributions $Y|(x, A = a)$ and $Y|(x, A \neq a)$. Differential fairness interprets disparate treatment of a classifier f on an individual with auditing features x as a non-negligible distance between $Y|(x, A = a)$ and $Y|(x, A \neq a)$.

Intention in Disparate Treatment Differential fairness is a useful definition of fairness in machine learning because it provides a test to whether the protected attribute A affects in a causal sense the classifier's outcome. This is important because, at least in the United States, there are legal precedents (see [1], [4], Title VII) that require a plaintiff to demonstrate the disparate treatment was intentional. Causality is defined here as the existence of path (either direct or indirect) between the protected attribute and the classifier's outcome that is not blocked by the auditing features x. Therefore, differential fairness sets a framework to establish causation of protected attributes on a classifier's outcome conditional on the auditing feature space.

2.3 Multi-differential fairness

Although useful, the notion of individual differential fairness suffers from one limitation: it cannot be computationally efficiently audited for. Looking for violations of individual differential fairness will require searching over a set of $2^{|\mathfrak{X}|}$ individuals. Moreover, if $\mathfrak{X}$ is rich enough empirically, a sample from a distribution over $\mathfrak{X} \times \mathfrak{A} \times \{-1, 1\}$ has a negligible probability to have two individuals with the same auditing feature x and different protected attributes a.

Therefore, we relax the definition of individual differential fairness and impose differential individual fairness for group of individuals or sub-populations. Formally, $\mathbb{C}$ denotes a collection of subsets S of $\mathfrak{X}$. The collection $\mathbb{C}_\alpha$ is α-strong if for $S \in \mathbb{C}$ and $y \in \{-1, 1\}$, $Pr[Y = y \,\&\, x \in S] \geq \alpha$.

Definition 2.2. *(Multi-Differential Fairness) Consider a α-strong collection $\mathbb{C}_\alpha$ of sub-populations of $\mathfrak{X}$. For $0 \leq \delta$, a classifier f is $(\mathbb{C}_\alpha, \delta)$-multi differential fair with respect to $\mathfrak{A}$ if for all protected attributes $a, a^{'} \in \mathfrak{A}$, $y \in \{-1, 1\}$ and for all $S \in \mathbb{C}_\alpha$:*

$$e^{-\delta} \leq \frac{Pr[Y = y|A = a, S]}{Pr[Y = y|A = a', S]} \leq e^{\delta} \tag{4}$$

Multi-differential fairness relaxes the notion of differential fairness by protecting sub-populations instead of individuals. Multi-differential fairness guarantees that the outcome of a classifier f is nearly mean-independent of protected attributes within any sub-population $S \in \mathbb{C}_\alpha$. The parameter δ controls for the amount of information related to protected attributes that the classifier leaks: smaller value of δ means smaller leakage. The fairness condition in Eq. 4 applies only to sub-populations with $Pr[Y = y \,\&\, x \in S] \geq \alpha$ for $y \in \{-1, 1\}$. This is to avoid trivial cases where $\{x \in S \,\&\, Y = y\}$ is a singleton for some y, which would imply that $\delta = \infty$.

Disparate Treatment versus Disparate Impact It is interesting to compare the definition of multi-differential fairness in 2.2 with previous definitions of fairness that are based on information leaked by the data about the protected attributes. First, if auditing features is empty, then multi-differential fairness devolves into a notion of disparate impact as in [14] or in [7]. Disparate impact is then a particular case of multi-differential fairness where no disparate treatment statement is made. On the other hand, if auditing features are all the features but protected attributes used to train f, multi-differential fairness is a framework to test whether there exists a sub-population for which there is a direct causal effect of protected attributes on the classifier's outcomes.

Collection of Indicators. The collection of sub-populations $\mathbb{C}$ can be equivalently thought as a family of indicators: for each $S \in \mathbb{C}$, there is an indicator $c_S : \mathfrak{X} \to \{-1, 1\}$ such that $c_S(x) = 1$ if and only if $x \in S$. The relaxation of differential fairness to a collection of groups or sub-population is akin to [22], [20] or [18] where $\mathbb{C}$ is the computational bound on the granularity of their definition of fairness. The richer $\mathbb{C}$, the stronger the fairness guarantee offers by definition 2.2. However, the complexity of $\mathbb{C}$ is limited by the fact that we identify a sub-population S via random samples drawn from a distribution over $\mathfrak{X} \times \mathfrak{A} \times \{-1, 1\}$. The rest of this paper shows that auditing for multi-differential fairness in polynomial time requires to limit the complexity of $\mathbb{C}$. Potential candidates for $\mathbb{C}$ will be the family short-decision trees or the set of conjunctions of constant number of boolean features. Therefore, auditing for multi-differential fairness will not check whether the fairness condition (4) holds for *all* sub-populations of $\mathfrak{X}$, but only check the fairness condition for all sub-populations that can be *efficiently identifiable.*

3 Fairness Diagnostics

The objective of our fairness diagnostic tool [**name**] is to find the sub-populations that are the most harmed by a classifier's outcomes. In the framework of multi-differential fairness it is equivalent to solve for $a \in \mathfrak{A}$

$$\sup_{S \in \mathbb{C}_\alpha} \ln\left(\frac{Pr[Y = 1|S, A = a]}{Pr[Y = 1|S, A \neq a]}\right). \tag{5}$$

Our approach succeeds at tackling three challenges: (i) if $\mathbb{C}$ is a rich and large class of subsets of the feature space $\mathfrak{X}$, an auditing algorithm linearly dependent on $|\mathbb{C}|$ can be prohibitively expensive; (ii) the data needs to be balanced conditional on protected attributes; (iii) finding efficiently the most-harmed sub-population implies that we can predict a function $c \in \mathbb{C}_\alpha$ for which we do not directly observe values $c(x)$.

3.1 Reduction to Agnostic Learning

Our first building block is to reduce the problem of certifying for the lack of differential fairness to a agnostic learning problem.

Multi Differential Fairness and Balanced Distribution The fairness condition 2.2 is unchanged if the feature distribution is reweighted, as long as the reweighting scheme does not depend on the classifier's outcome Y. More formally, for any weights $u : \mathfrak{X} \times \mathfrak{A} \to \mathbb{R}$ such that $u(x, a) > 0$ and $E[u] = 1$,

$$Pr[Y|A = a, S] \leq e^\delta Pr[Y|A = a^{'}, S] \iff Pr_u[Y|A = a, S] \leq e^\delta Pr_u[Y|A = a^{'}, S], \tag{6}$$

the sub-script u indicating that the probabilities are taken over the reweighted distribution.

Suppose that for any $a \in \mathfrak{A}$, we have oracle access to the importance sampling weight $w_a(x) = \frac{1-P[A=a|x]}{P[A=a|x]}$. For any distribution D_f over $\mathfrak{X} \times \mathfrak{A} \times \{-1, 1\}$ denote D_f^w the corresponding balanced distribution. Note that once reweighted by w_a, for any sub-population $S \in \mathbb{C}$, auditing features does not reveal anything about the distribution of the protected attribute A: $Pr_w[A = a|X, S] = Pr_w[A \neq a|X, S]$. With a balanced distribution, the multi differential fairness condition can be rewritten as follows: for all protected attributes $a \in \mathfrak{A}$, $y \in \{-1, 1\}$ and for all $S \in \mathbb{C}_\alpha$

$$Pr_w[A = a|S, y] \leq \frac{e^\delta}{e^\delta + 1}, \tag{7}$$

where the sub-script w reminds that the distribution D_f^w is balanced. Since the distribution D_f^w induced by f is balanced, auditing features x do not reveal any information on protected attributes and multi-differential fairness can then be interpreted as an upper bound on ability to predict A given the classifier's outcome for any sub-population $S \in \mathbb{C}$ with $Pr_w[S, y] \geq \alpha$ for $y \in \{-1, 1\}$. A violation of $(\mathbb{C}_\alpha, \delta)$- multi differential fairness is a sub-population $S \in \mathbb{C}_\alpha$ such that

$$Pr_w[A = a|S, y] - \frac{1}{2} \geq \frac{e^\delta}{e^\delta + 1} - \frac{1}{2}. \tag{8}$$

Therefore, a γ-unfairness certificate is a subset $S \in \mathbb{C}$ such that there exists $y \in \{-1, 1\}$ with

$$Pr_w[S, y] \left\{ Pr_w[A = a|S, y] - \frac{1}{2} \right\} \geq \gamma, \tag{9}$$

with $\gamma = \alpha \left(e^\delta/(1 + e^\delta) - 1/2 \right)$. γ is then a measure of multi-differential unfairness that combines the size of the sub-population where a violation exists and the magnitude of the violation. With balanced distribution, certifying multi-differential fairness is akin to searching for γ-unfairness certificate.

Definition 3.1. *(Certifying Multi-Differential Fairness). Let $\gamma, \epsilon > 0$, $\eta \in (0, 1)$ and $\mathbb{C}_\alpha$ be an α-strong collection of sub-populations in $\mathfrak{X}$. An (ϵ, η)- certifying algorithm $M(\epsilon, \delta)$ is an algorithm that for any sample from a distribution D_f induced by a classifier over $\mathfrak{X} \times \mathfrak{A} \times \{-1, 1\}$, outputs a $\gamma - \epsilon$-unfairness certificate with probability $1 - \eta$ whenever f is γ-multi differential unfair; and, certifies fairness with probability $1 - \eta$ whenever f is is γ-multi differential fair.*

Moreover, $M(\epsilon, \delta)$ is an efficient certifying algorithm if it requires poly$(\log(|\mathbb{C}_\alpha|, \log(1/\eta), 1/\epsilon))$ samples and runs in poly$(\log(|\mathbb{C}_\alpha|, \log(1/\eta), 1/\epsilon))$.

Searching for γ-unfairness certificate can be formulated as a problem of detecting correlations since it reduces to predict either $\{a_i y_i\}$ given $\{x_i\}$.

Lemma 3.1. *Let $a \in \mathfrak{A}$. Suppose oracle access to importance sampling weight w_a. f is $\gamma-$ multi-differential unfair if and only there exists $c \in \mathbb{C}$ such that either $Pr_w[AY = c] \geq 1 - \rho_+ + 4\gamma$ or $Pr_w[A(\neg Y) = c] \geq 1 - \rho_- + 4\gamma$, where $\rho_s = Pr[A = sY]$ with $s = \pm$ and $\neg Y = -Y$.*

To gain intuition for lemma 3.1, suppose that we are looking for violation of multi-differential fairness with $a = 1$ and $y = 1$ in Eq. (9). Since the distribution D_f^w is balanced, a violation will be a sub-population where conditional on $Y = 1$, more than half the time protected attributes and Y agree ($A = Y = 1$); and, conditional on $Y = -1$, more than half the time $A = Y = -1$. Therefore, certifying the lack of multi-differential fairness is equivalent to find $c \in \mathbb{C}_\alpha$ that predicts where A and Y agrees, that is $c = 1$ whenever $AY = 1$ and $c = -1$ whenever $AY = -1$.

The equivalence in lemma 3.1 is useful to rephrase the certifying problem into an agnostic learning problem. A concept class $\mathbb{C}$ is agnostically efficiently learnable if and only if for all $\epsilon, \eta > 0$, there exists an algorithm $\mathfrak{M}$ that given access to a distribution $\{x_i, o_i\} \sim D \times \{-1, 1\}$ outputs with probability $1 - \eta$ in $poly(\log(|\mathfrak{C}, \log(\frac{1}{\eta}), \frac{1}{\epsilon})$ outputs a function $h \in \mathbb{C}$ such that

$$Pr_D[g = h] + \epsilon \geq max_{c \in \mathbb{C}} Pr_D[g = c].$$

We show that if the collection of subpopulation $\mathbb{C}$ admits an efficient agnostic learner, we could use that learner to construct an algorithm certifying multi-differential fairness.

Theorem 3.2. *Let $\epsilon > 0$, $\beta > 0$ and $\mathbb{C} \subset 2^{\mathfrak{X}}$. Suppose oracle access to importance sampling weight w_a for any $a \in \mathfrak{A}$. There exists an efficient (ϵ, η)-auditing algorithm for $\mathbb{C}$ on balanced distributions if and only if $\mathbb{C}$ admits a (ϵ, η) efficient agnostic learner for any balanced distribution over $\mathfrak{A}$.*

The result in theorem 3.2 makes clear that not all sub-population can be efficiently audited for multi-differential fairness. There are many concept classes $\mathbb{C}$ for which agnostic learning is a NP-hard problem, including for any learning methods that outputs a half-space as an hypothesis (see [15]). However, there are classes for which efficient agnostic learners exist (see [21]).

Based on theorem 3.2 and its proof, we convert the certifying algorithm problem into the following empirical loss minimization: for a sample $\{(x_i, a_i), y_i)\}_{i=1}^m \sim D_f^w$, solve

$$opt = \min_{c \in \mathbb{C}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\left(a_i y_i \neq c(x_i)\right) \tag{10}$$

and then compute γ as:

$$\gamma = \max_{s=\pm 1} \frac{opt + \widehat{\rho}_s - 1}{4}, \tag{11}$$

where $\widehat{\rho}_s$ is the sample estimate of $Pr_w[A = sY]$ for $s = \pm 1$.

3.2 Imbalanced Data

The risk minimization problem Eq. (10) allows to certify efficiently the lack of multi-differential fairness. However, in theorem 3.2, we assume oracle access to importance sampling weights w. However, most of the time, importance sampling weights are unobserved and need to be estimated. Moreover, variance of those estimates are known to be large (see [9] or). In this section, we propose to obtain a balanced representation technique that allows to certify multi-differential fairness without the need to estimate importance sampling weights.

Imbalance Problem With imbalanced distributions, the multi-differential fairness condition in Eq. (7) includes the term $w_S = Pr[A = a|S]/Pr[A \neq a|S]$ for any $a \in \mathfrak{A}$:

$$Pr_w[A = a|S, y] \leq \frac{e^\delta w_S}{e^\delta w_S + 1}, \tag{12}$$

Therefore, solving the risk minimization problem Eq. (10), when applied to imbalanced distribution, cannot distinguish the case of a high value of δ from the case of low value δ but a high value of w_S. The latter situation is the result of imbalances in the data that result from social, cultural or historical discriminations encoded in the data; the former is an issue with the classifier f itself that needs to be audited for.

One approach is to obtain the importance sampling weights directly by estimating the density $P[A = a|x]$. This method is used in propensity-score matching methods (see [23] and [16] for this plug-in approach) in the context of counterfactual analysis. However, as noticed in [9], exact or estimated importance sampling result in large variance in finite sample. In fact, estimating the distribution $P[A = a|x]$ to obtain the weight $w_a(x)$ may be an overkill. Instead, we follow [17] and [10] and observe that if u is a weight function,

$$Pr_w[AY \neq c] \leq Pr_u[AY \neq c(\phi)] + MMD^{\phi}(uP_a, P_{\neg a}) \tag{13}$$

with equality whenever $u = w_a$. MMD is the linear maximum mean discrepancy between the distribution of features $p(x, A = a)$ weighted by u and the distribution of $p(x, A = \neg a)$:

$$MMD^{\phi}(uP_a, P_{\neg a}) = \left\| E\left[\frac{1}{n_a}\sum_{i,A=a} u(\phi(x_i))\phi(x_i) - \frac{1}{n_{a'}}\sum_{i,A=a'} \phi(x_i)\right]\right\|^2 \tag{14}$$

where $\phi : \mathfrak{X} \rightarrow \mathfrak{F}$ is any feature map into a feature space $\mathfrak{F}$ (see [17]). In other words, the linear maximum mean discrepancy between $uP(., A = a)$ and $P(., A \neq a)$ is measured once the features are transported into the space $\phi(\mathfrak{X})$.

Therefore to certify for multi-differential fairness, our approach consists into finding $c \in \mathbb{C}$, ϕ and u that minimizes the empirical counterpart of the upper bound in (14):

$$\begin{aligned} L(w, c, \phi) =& \frac{1}{N}\sum_i u_i(\phi(x_i))\mathbb{1}(c(\phi(x_i)) \neq a_i y_i) + Reg(c) \\ &+ \left\| \frac{1}{n_a}\sum_{i,A=a} u_i(\phi(x_i))\phi(x_i) - \frac{1}{N - n_a}\sum_{i,A\neq a} \phi(x_i)\right\|^2 + Reg(u) \end{aligned} . \tag{15}$$

In our implementation MMD_{NET}, the feature representation ϕ is learned via a neural network that is then shared with both tasks of minimizing the re-weighted certifying risk and the distributional shift between $uP(., A = a)$ and $P(., A \neq a)$.

Generic uniform convergence argument allows to derive the sample complexity and correctness of our certifying algorithm 1.

Theorem 3.3. *(Sample Complexity and Correctness of Algorithm 1)Let $\epsilon > 0$ and $\eta \in (0, 1)$. Suppose that $\mathbb{C}$ is a concept class of dimension $d(\mathbb{C}) < \infty$. Algorithm 1 is $(\epsilon, \eta)-$ certifying algorithm for samples of size $m \geq m(\epsilon, \eta, d)$, where*

$$m =$$

3.3 Unfairness Diagnostics: Worst-Case Violation

Algorithm 1 presented above allows to certify whether any black box classifier is multi-differential fair with only $O(\log(|\mathbb{C}|)$ samples. However, it does not identify the sub-population in $\mathbb{C}_\alpha$ with the strongest violation of multi differential fairness (i.e., with the largest value δ in Eq. (2.2)). This is because algorithm 1 does not distinguish a large sub-population S with low value of δ from a smaller sub-population with larger value of δ. Finding the strongest violation is useful to (i) diagnostic the source of multi-differential unfairness of a classifier; and, (ii) identify the individuals which could be the most harmed by a classifier's outcome.

Algorithm 1 Certifying Fairness Algorithm (CFA)

```
**Input:** $\{((x_i, a_i), y_i)\}_{i=1}^m$, $\mathbb{C} \subset 2^{|\mathfrak{X}|}$, $\lambda_u$, $\lambda_c$.
$\hat{\rho}_+ \leftarrow \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(a_i = y_i)$
$\hat{\rho}_- \leftarrow \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(a_i = -y_i)$
$u^*, \phi^* = argmin \left\| \frac{1}{n_a} \sum_{i,A=a} u_i \phi(x_i) - \frac{1}{N - n_a} \sum_{i,A \neq a} \phi(x_i) \right\|^2 + \lambda_u \|u\|^2$
$c^* = argmin_{c \in \mathbb{C}} \frac{1}{m} \sum_{i=1}^{m} u(x) \mathbb{1}\left(a_i y_i = c(\phi(x_i))\right) + \lambda_c Reg(c)$
$\hat{\gamma} a \leftarrow \max\{\frac{opt+1-\hat{\rho}_+}{4}, \frac{opt+1-\hat{\rho}_-}{4}\}$
**Return** $\hat{\gamma}-$ unfair
```

Worst-Case Violation Problem The objective is to identify for any $a \in \mathfrak{A}$ the sub-population S in $\mathbb{C}$ that solves:

$$\delta_m \equiv \sup_{S \in \mathbb{C}} \ln \left(\frac{Pr[Y = 1|S, A = a]}{Pr[Y = 1|S, A \neq a]} \right). \tag{16}$$

To illustrate the challenges of the worst-case violation problem, consider a binary protected attributes $A = \{-1, 1\}$ where there exist $S_0, S_\delta \in \mathbb{C}$ with no violation of multi differential fairness in sub-population S_0, but a $\delta-$ multi differential fairness violation in S_δ (e.g. $P[Y|A = 1, S_\delta] < e^\delta P[Y|A = -1, S_\delta]$). Consider $c, c^{'} \in \mathbb{C}$ such that $c(x) = 1$ if and only if $x \in S_\delta$ and $c^{'}(x) = 1$ if and only if $x \in S_\delta \cup S_0$. Both c and $c^{'}$ have the same accuracy on $S_\delta \cup S_0$ once the distribution is rebalanced. Therefore, algorithm 1 will pick indifferently c or $c^{'}$ as unfairness certificate, although $c^{'}$ mixes the worst-case violation S_δ with a sub-population without any violation of differential fairness.

Worst-Case Violation Algorithm (WVA) At issue in the previous example is that for the sub-population S_0 with no violation of multi differential fairness, choosing $c = 1$ or $c = -1$ will lead to the same empirical risk used in our certifying procedure CFA. Our worst-case violation algorithm WVA (see 2) puts a slightly larger weight on samples $((x_i, a_i), y_i)$ whenever $a_i y_i \neq 1$ so that the empirical risk is now smaller if $c = -1$ wherever there is no violation of multi differential fairness. More generally, our worst violation algorithm is an iterative procedure that at each iteration t, increases by $1 + \xi t$ the weight on samples $((x_i, a_i), y_i)$ whenever $a_i y_i \neq 1$, where $\xi > 0$. At iteration t, the solution c_t^* of the empirical risk minimization (10) identifies a sub-population $S_t = \{x | c_t^*(x) = 1\}$ for which $\delta \geq \ln((1 - h(\xi t))/h(\xi t))$, where h is an increasing function. The algorithm 2 terminates whenever either $|S_t| \leq \alpha$ or $c_t^*(x) = -1$ for all x. At the last iteration before termination, theorem 3.4 shows that algorithm 2 will identify a sub-population c^* with a $\widehat{\delta}(c^*)$-multi differential fairness violation and $\widehat{\delta}(c^*)$ asymptotically close to δ_m.

Theorem 3.4. *Suppose $\xi > 0, \epsilon > 0, \eta \in (0, 1)$ and $\mathbb{C} \subset 2^{\mathfrak{X}}$ is $\alpha-$strong. Suppose that the classifier f has been certified with γ-multi-differential unfairness. Denote δ_m the worst-case violation of multi differential fairness for $\mathbb{C}$ as defined in* (16)*. With probability $1 - \eta$, with $O\left(\frac{1}{\epsilon^2} \log(|C|) \log\left(\frac{4}{\eta}\right)\right)$*

Algorithm 2 Worst Violation Algorithm (WVA)

```
**Input:** $\{((x_i, a_i), y_i)\}_{i=1}^m$, $\mathbb{C} \subset 2^{|\mathfrak{X}|}$, $\xi$, $\alpha$, balancing weights $u$
$\delta_{-1} = 0$; $\delta_0 = 1$, $\alpha_0 = 1$
**while** $\delta_t >= \delta_{t-1}$ or $\alpha_t > \alpha$ **do**
    **for** i=1..m **do** $u_{it} \leftarrow u_i(1 + \xi t)$ if $a_i y_i = -1$
    $c^* = argmin_{c \in \mathbb{C}} \frac{1}{m} \sum_{i=1}^{m} u_{it}(x) \mathbb{1}\left(a_i y_i = c(\phi(x_i))\right) + \lambda_c Reg(c)$
    $\hat{\delta}_t \leftarrow \frac{\sum_{i=1, c(x_i=1)}^{m} u_i(x_i) \mathbb{1}(y_i = 1) \mathbb{1}(a_i = 1)}{\sum_{i=1, c(x_i=1)}^{m} u_i(x_i) \mathbb{1}(y_i = 1) \mathbb{1}(a_i = -1)}$
    $\hat{\alpha}_t \leftarrow \frac{\sum_{i=1)}^{m} u_i(x_i) \mathbb{1}(c_i(x_i) = 1)}{\sum_{i=1}^{m} u_i(x_i)}$
    $t \leftarrow t + 1$
**Return** $\ln(\delta_t)$.
```

samples and after $O\left(\frac{4(\gamma+\alpha)}{2\gamma+3\alpha} \frac{2(4\gamma - 2\rho(y)+1)}{\xi}\right)$ *iterations, algorithm 2 learns* $c \in \mathbb{C}$ *such that*

$$\left| \ln\left(\frac{Pr_w[Y = 1 | A = 1, c(x) = 1]}{Pr_w[Y = 1 | A \neq a, c(x) = 1]} \right) - \delta_m \right| \leq \epsilon. \tag{17}$$

The number of iterations of our worst-case violation algorithm in 2 can be bounded above by a constant $O(1)$ by choosing $\xi = O\left(\frac{4(\gamma+\alpha)}{2\gamma+3\alpha}(4\gamma - 2\rho(y) + 1)\right)$. [One remark on the weights u.]

3.4 [name]

Putting the building blocks together allows us to design a fairness diagnostic tool [**name**] that identifies efficiently and with statistical confidence the victims of differential unfairness.

Architecture Inputs are a dataset with a classifier's outcomes (label ±1) along with auditing features. [**name**] first uses a neural network MMD with four fully connected layers of 36 neurons to obtain weights u that minimizes the maximum-mean discrepancy function. The outputs of the last layer in the neural network are used as a representation of the feature space and serve as an input to the CFA component along with the estimated weights u. The CFA component produces a certificate c of unfairness. If the size of the corresponding sub-population is larger that the target α, the noise infuser component re-adjusts the weights as in 2 and the CFA block is rerun. This iterative procedure stops when there is no increase in the estimated differential unfairness parameter δ or the identified worst-case violation has a size smaller than α. When terminating, [**name**] outputs an estimate of the most-harmed sub-population c along with an estimate of δ_m.

Cross-Validation The auditor decides ex-ante on the minimum size α of the worst-case violation he would like to identify. Smaller α provides more granularity but decreases the statistical confidence for the estimated value δ_m. Regularization parameters in both the MMD and CFA components need to be fine-tuned. The advantage of our approach is that, although we do not have ground truth for unfair treatment, we can propose heuristics to cross-validate our choice of regularization parameters. First, we split 70%/30% the input data into a train and test set. Using a $5-$fold cross-validation, [**name**] is trained on four folds and a grid search looks for regularization parameters that minimize the maximum-mean-discrepancy (in MMD) and the empirical risk (in CFA) on the fifth fold. Once [**name**] is trained, the estimated δ_m and the corresponding characteristics of the most-harmed sub-population is computed on the test set.

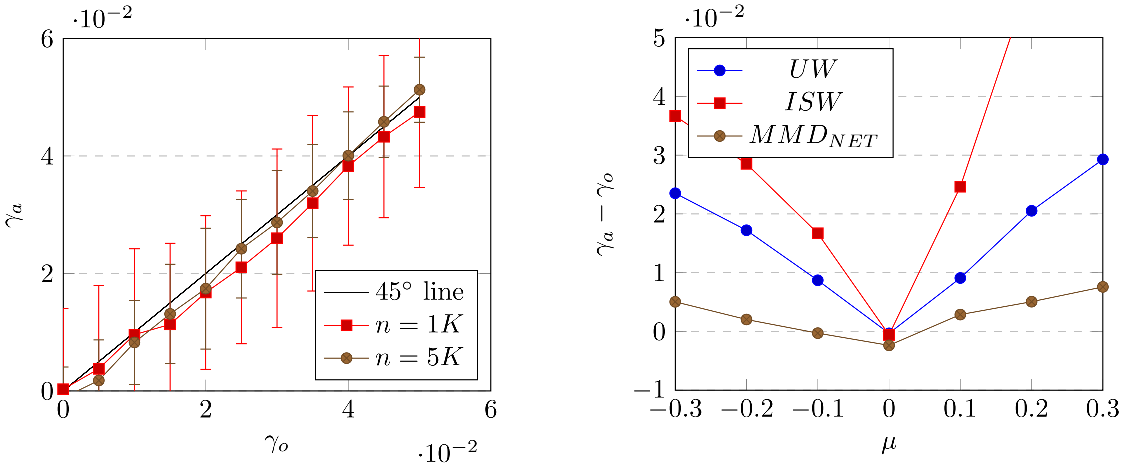
4 Experiments

4.1 Synthetic Data

A synthetic data is constructed by drawing independently two features X_1 and X_2 from two normal distribution $N(0,1)$. We consider a binary protected attribute $\mathfrak{A} = \{-1, 1\}$ drawn from Bernouilli distribution with $A = 1$ obtained with probability $w(x) = \frac{e^{\mu*(x_1-x_2))^2}}{1+e^{\mu*(x_1+x_2))^2}}$ and $A = -1$ with probability $1 - w(x)$ for $x = (x_1, x_2)$. μ is an unbalance factor. $\mu = 0$ means that the data is perfectly balanced with $w(x) = \frac{1}{2}$. As μ increases, the distribution $Pr(x|A = 1)$ becomes more dense than $Pr(x|A = -1)$ in the for points away from the 45° diagonal in a (x_1, x_2) plane. The data is labeled according to the sign of $(X_1 + X_2 + e)^3$, where is e is a noise drawn from $N(0, 0.2)$. The audited classifier f is a logistic regression classifier that is altered to generate instances of differential unfairness: with probability $1 - \nu \in [0, 1)$, the sign of the classifier's outcomes Y from individuals with $A = -1$ is changed from -1 to $+1$ if $x_1^2 + x_2^2 \leq 1$; if $A = 1$, all classifier's outcomes are changed from -1 to $+1$ if $x_1^2 + x_2^2 \leq 1$. For $\nu = 0$, the audited classifier is differentially fair; however, as ν increases, the half circle $\{(x_1, x_2)|x_1^2 + x_2^2 \leq 1 \text{ and } y = -1\}$ there is a fraction ν of individuals with protected attribute equal to -1 who are not treated similarly as individuals with protected value equal to 1.

Sample Complexity The auditing algorithm MMD_{NET} is trained using a decision tree and a unbalanced data ($\mu = 0.2$). Given our setting, for any value of ν we can compute the actual level of differential unfairness γ_o and compare it to the value γ_a estimated by the auditing algorithmMMD_{NET}. Figure 2a plots γ_a against γ_o after 100 simulations for value of β varying from 0 to 0.5 and fixed sub-population size $\alpha = 0.1$. The experiment is conducted with data set of size $n \in \{1000, 5000\}$. The estimated unfairness level γ_a is unbiased since the plots nearly align with 45° line. Larger sample ($5K$) have little effect on how bias the auditor MMD_{NET} is, but reduces the variance of the estimated γ_a.

Unbalanced Data To test the performance of our certifying algorithm on unbalanced data, we repeat the previous experiment with μ varying from -0.3 to 0.3. We compare the performance of three reweighting approaches: uniform weight (UW); direct use of the importance sampling weights w (ISW); . Figure 2b shows that absent of a reweighting scheme (UW) the certifying algorithm fails to estimate correctly the classifier's unfairness if $\mu \neq 0$. This is in line with the observation made in section 3 that the auditing algorithm needs to control for the information leaked by the auditing features x when measuring the additional leakage from the classifier's outcome y. Our preferred



(a) Effect of unfairness intensity β on auditor's performance.

(b) Effect of unbalanced data on auditor's performance.

Figure 2: Certifying γ- multi differential unfairness. The auditor is a decision tree whose depth is tuned using a 5-fold cross validation. The weights function u is obtained using a neural network with four hidden layers with eight neurons each. The plots show the average of 100 experiments. Figure 2a auditor's bias when estimating γ_a is measured by deviations from the 45° line; the unbalance parameter μ is set to 0; the size α of sub-population with a fairness violation is set to 0.1; and, the intensity β of the fairness violation varies from 0 to 0.5. Figure 2b: bias is measured as the difference $\gamma_a - \gamma_o$; the size α of sub-population with a fairness violation is set to 0.1; the intensity β of the fairness violation is set to 0.5; the balancing parameter μ varies from -0.3 to 0.3.

method MMD_{NET} performs well as there is little bias in the estimate γ_a even at large value of the unbalancing factor ν. Note using importance sampling weights directly does worse than a uniform sampling: this confirms previous observations in the literature that in finite sample, the variance of the importance sample weights can affect be detrimental to a re-balancing approach.

Concept Class Figure 3 runs a similar experiment but varies with decision trees of depth varying from 5 to 40. At small level of unfairness, a less complex concept class $\mathbb{C}$ generates less bias in the estimate of γ. A shorter decision tree out-performs significantly deeper structures. This result, in line with the sample complexity presented in theorem 3.3 justifies the concept of multi fairness: in order to be statistically meaningful, the granularity of the sub-populations for which multi differential fairness is audited for is bounded by the complexity of $\mathbb{C}$.

Worst Violations As noted in section 3, the certifying algorithm does not allow to distinguish between a very intense (high δ) on a small sub-population (small α) from a less intense violation (small δ) on a large sub-population (large α). We test whether algroithm 2 is able to identify that in the synthetic data, the differential fairness violation occurs in the sub-space $\{(x_1, x_2)|x_1^2 + x_2^2 \leq 1 \text{ and } y = -1\}$. Figure **??** shows that there is little bias in the estimate of δ_a for intermediate values of δ. The algorithm underestimates the true value of δ_o for large violations of differential fairness.

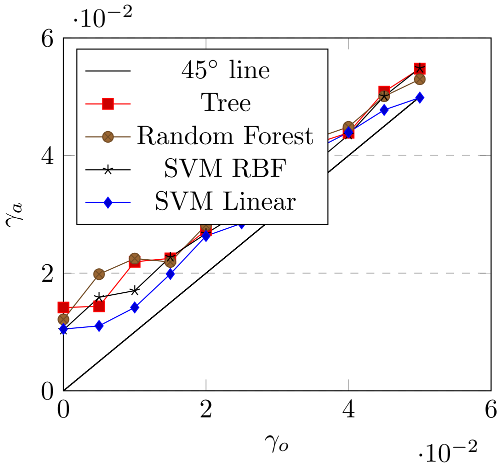


Figure 3: Effect of auditor's class on auditor's performance.

Figure 4: Certifying γ- multi differential unfairness. The auditor is a decision tree whose depth is tuned using a 5-fold cross validation. The weights function u is obtained using a neural network with four hidden layers with eight neurons each. The plots show the average of 100 experiments. Figure 3 auditor's bias when estimating γ_a is measured by deviations from the 45° line; the unbalance parameter μ is set to 0.2; the size α of sub-population with a fairness violation is set to 0.1; and, the intensity β of the fairness violation varies from 0 to 0.5.

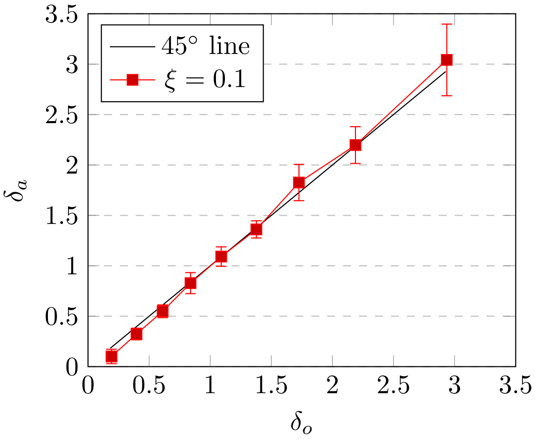


Figure 5: Finding δ for the worst violation of multi-differential fairness. The auditor is a RBF support vector machine whose regularization is tuned using a 5-fold cross validation. The weights function u is obtained using a neural network with four hidden layers with eight neurons each. The plots show the average of 100 experiments. auditor's bias when estimating γ_a is measured by deviations from the 45° line; the unbalance parameter μ is set to 0.2; the size α of sub-population with a fairness violation is set to 0.1; and, the intensity β of the fairness violation is set to 0.5. The bias in the estimated δ is measured as the difference $\delta_a - \delta_o$

4.2 Case Study: COMPAS

We chose to apply first our method to the COMPAS algorithm because it is a widely used to assess the likelihood of a defendant to become a recidivist. Records from Broward County, Florida are publicly available and expansive analysis of the COMPAS fairness have been already carried out. Our novelty is to apply our multi-differential fairness framework and explore whether there exists in those records group of individuals that could argue for a disparate treatment akin to an "intentional" discrimination.

Data Description The data collected by ProPublica in Broward County from 2013 to 2015 contain $7K$ individuals along with a risk score and a risk category assigned by COMPAS. We transform the risk category into a binary variable equal to 1 for individuals assigned in the high risk category (risk score between 8 and 10). The data provides us with information related to the historical criminal history, misdemeanors, gender, age and race of each individual.

Certifying the Lack of Differential Fairness First, we assess whether the COMPAS risk classification is multi-differential fair. The data is randomly split into train and test sets using 70%/30% ratio. The auditor uses a four layers fully connected neural network to re-balance the data and a random forest to certify differential fairness. The assessment is made for a binary protected attribute: whether an individual self-identified as Afro-American. We find a significant level of differential unfairness in the COMPAS risk classification (see Table **??**) when the binary protected attribute is whether an individual self-identified as Afro-American. The unfairness is slightly stronger with a auditing feature space limited to the count of prior felonies and the degree of the current charge.On the other hand, we do not find any evidence that the COMPAS risk classification violates differential fairness when the protected attributes is a binary variable indicating whether an individual self-identifies a Male.

Features	Race	Gender
I, II, III, IV, V	0.023761 ± 0.004312	-0.0001 ± 0.0006
I, II, III	0.023599 ± 0.00384	-0.000037 ± 0.0002
I, II	0.027382 ± 0.019383	-0.00005 ± 0.0001

Table 1: Certifying the lack of differential fairness in COMPAS risk classification. Features are as follows: I: count of prior felonies; II: degree of current charge (criminal vs non-criminal); III: age; IV: count of juvenile prior felonies; V: count of juvenile prior misdemeanors. Standard deviations are obtained by drawing 100 train/test splits. In the first column, Race, the protected attribute is a binary variable indicating whether an individual is self-identified as Afro-American; in the second column, Gender, the protected attribute is a binary variable indicating whether an individual is self-identified as Male

Worst Violations The results in Table **??** do not allow to distinguish the case of a small violation of differential fairness spread evenly across the whole feature distribution from the case of large violations concentrated on small sub-populations. We run our worst-case violation algorithm on 100 0.7/0.3 train/test split and average the characteristics of the sub-population over the 100 experiments. Table 2 shows that violation of differential fairness is more severely concentrated on African American individuals who have a small or non-existent criminal and misdemeanor history. In the sub-population with the worst-case violation, African American individuals have a similar criminal history as non-African American individuals, but are three times more likely to be classified as high risk. Note that although individuals self-identified as African American do not have the same criminal history than the rest of the population (see the two first columns of table 2), our worst-case violation algorithm is able to extract effectively a sub-population where African-American individuals are similar to the rest of the group, but are treated differently. Note also that an African American in the worst-case violation sub-population, albeit with no criminal history, is more likely to be classified as high risk than a non-African American, with an average more felonies and misdemeanors, in the overall population.

Variable	Population		Worst-case Violation	
	African-American	Other	African-American	Other
Prior Felonies	4.44 ± 5.58	2.46± 3.76	0.79± 0.24	0.67 ± 0.17
Charge Degree	0.31 ± 0.46	0.4 ± 0.49	0.74 ± 0.23	0.74 ± 0.2
Juvenile Felonies	0.1 ± 0.49	0.03± 0.32	0.01± 0.02	0.0 ± 0.02
Juvenile Misdemeanor	0.14 ± 0.61	0.04± 0.3	0.01± 0.02	0.01 ± 0.01
High Risk	0.14 ± 0.35	0.05± 0.22	0.06± 0.04	0.02 ±0.01

Table 2: Identifying the worst-case violation of differential fairness in the COMPAS risk score. The protected attribute is whether the individual is self-identified as African American. The worst-case violation algorithm is run with a random forest of 100 tree stumps of depth 2 and an incremental weight increase equal to 0.1. The algorithm is run 100 times, each time with a different 0.7/0.3 train/test set split. Estimates in the tables are obtained from the test set.

Disparate Treatment Pick up Alice, Bob, Carol and Dick who belongs to sub-population for which a disparate treatment can be shown.

4.3 Other Datasets: Group Fairness vs. Multi-Differential Fairness

5 Appendix

5.1 Analysis of Algorithm 2

Let $a \in \mathfrak{A}$. Assume that the classifier f is γ-unfair. Let δ_m denote the worst-case violation. Note that by definition of γ and δ_m:

$$\gamma = \alpha \left(\frac{e^{\delta_m}}{e^{\delta_m} + 1} - \frac{1}{2} \right) \tag{18}$$

At each iteration t, denote c_t^* the solution of the following optimization problem

$$max_{c \in \mathbb{C}} E_{D_f^w} \left[\sum_{i=1}^{m} w_{it} \mathbb{1}_a \left(a_i y_i = c(x_i) \right) \right], \tag{19}$$

where w_{it} are the weights at iteration t and the expectation is taken over all the samples of size m drawn from D_f^w.

$$w_{it} = \begin{cases} w_i(1 + \nu t) \text{ if } y_i \neq a_i \\ w_i \text{ otherwise.} \end{cases} \tag{20}$$

Note first that

$$
\begin{aligned}
E_{D_f^w}\left[\sum_{i=1}^{m} w_{it}a_iy_ic_t(x_i)\right] &= E_D\left[\sum_{i=1,a_i=y_i}^{m} w_ic_t(x_i) - \sum_{i=1,a_i\neq y_i}^{m} w_ic_t(x_i) - \xi t \sum_{i=1,a_i\neq y_i}^{m} w_ic_t(x_i)\right] \\
&= 2 * Pr_w[c_t(x_i) = a_iy_i)] - 1 - \xi t E_{D_f^w}\left[\sum_{i=1,a_i\neq y_i}^{m} w_ic_t(x_i)\right]
\end{aligned}
\tag{21}
$$

On the other hand,if c_o denotes the indicator such that $c_o(x) = -1$ for all $x \in \mathfrak{X}$,

$$
E_D\left[\sum_{i=1}^{m} w_{it}a_iy_ic_0(x_i)\right] = 2 * Pr_w[a_i \neq y_i] - 1 + \xi t E_{D_f^w}\left[\sum_{i=1,a_i\neq y_i}^{m} w_i\right]. \tag{22}
$$

Therefore, either $c_t = c_o$ or

$$
Pr_w[c_t(x_i) = a_iy_i)] \geq Pr_w[a_i \neq y_i] + \frac{1}{2}\xi t E_{D_f^w}\left[\sum_{i=1,a_i\neq y_i,c_t(x_i)=1)}^{m} w_ic_t(x_i)\right]. \tag{23}
$$

Note that

$$
E_{D_f^w}\left[\sum_{i=1,a_i\neq y_i,c_t(x_i)=1)}^{m} w_ic_t(x_i)\right] = Pr_w[a_iy_i \neq c_t(x_i)|c_t(x_i) = 1]. \tag{24}
$$

It follows that if $c_t \neq c_o$,

$$
Pr_w[AY = c_t|c_t = 1] \geq 1 - \frac{2}{\xi t}\left(Pr_w[c_t = AY)] - \rho(y)\right), \tag{25}
$$

where $\rho(y) = Pr_w[A \neq Y]$. Since the classifier f is γ-unfair, we know that

$$
\max_{c \in \mathbb{C}} Pr_w[AY = c] = 4\gamma + 1 - \rho(y). \tag{26}
$$

Therefore, since $c_t \in \mathbb{C}$, at iteration t, either $c = c_o$ or

$$
Pr_w[AY = c_t|c_t = 1] \geq 1 - \frac{2(4\gamma + 1 - 2\rho(y))}{\xi t} = 1 - h(\xi t), \tag{27}
$$

where $h(\xi t) = \frac{2(4\gamma+1-2\rho(y))}{\xi t}$. Therefore $Pr_w[Y = 1|c_t = 1, A = 1] \geq 1 - h(\xi t)$ or $Pr_w[Y = -1|c_t = 1, A = -1] \geq 1 - h(\xi t)$. Without loss of generality, we can assume $Pr_w[Y = 1|c_t = 1, A = 1] \geq 1 - h(\xi t)$. Let $\delta(c_t) = \ln\left(\frac{Pr_w[Y=1|A=1,c=1])}{Pr_w[Y=1|A=-1,c=1]}\right)$. It follows that whenever $c_t \neq c_o$

$$
\frac{e^{\delta(c_t)}}{1 + e^{\delta(c_t)}} \geq 1 - h(\xi t). \tag{28}
$$

Therefore,

$$
\delta_m \geq \delta(c_t) \geq \ln\left(\frac{1 - h(\xi t)}{h(\xi t)}\right). \tag{29}
$$

What matters from (29) is that since h is a decreasing function of t, there exists T such that $\delta(c_T) = \delta_m$:

$$T = \frac{2(4\gamma + 1 - 2\rho(y))}{\xi}\left(e^{\delta_m} + 1\right) \tag{30}$$

We are guaranteed that the algorithm will not stop later than T since by definition of $\delta(c_T)$ and γ:

$$\begin{aligned} Pr_w[Y = 1, c_T = 1]\left(\frac{e^{\delta(c_T)}}{1 + e^{\delta(c_T)}} - \frac{1}{2}\right) &= Pr_w[Y = 1, c = 1]\left(Pr_w[A = 1|Y = 1, c = 1] - \frac{1}{2}\right) \\ &\leq \gamma = \alpha\left(\frac{e^{\delta_m}}{e^{\delta_m} + 1} - \frac{1}{2}\right), \end{aligned} \tag{31}$$

which implies $Pr_w[Y = 1, c_T = 1] \leq \alpha$. Therefore the size of the sub-population identified with a fairness violation by c_T will be at most α and the algorithm will stop at the latest T.

It remains to show that when the algorithm stops (i.e $Pr_w[c_t = 1, Y = 1] = \alpha$), $\delta(c_T)$ is δ_m. p To do so, note that for c_t to solve the optimization problem (19), it has to be chosen over c_m, the sub-population in $\mathbb{C}$ that corresponds to the worst-case violations. This implies that

$$\begin{aligned} 2Pr_w[AY = c_t] - 2Pr_w[AY = c_m] &\geq \xi t E_{D_f^w}\left[\sum_{i=1, a_i \neq y_i}^{m} w_i(c_t(x_i) - c_m(x_i))\right] \\ &= 2\xi t\left(Pr_w[c_t = 1|A \neq Y] - Pr_w[c_m = 1|A \neq Y]\right) \\ &= 2\frac{\xi t}{Pr_w[A \neq Y]}\left(Pr_w[AY \neq c_t|c_t = 1]Pr_w[c_t = 1]\right. \\ &\quad \left. - Pr_w[AY \neq c_m|c_m = 1]Pr_w[c_m = 1]\right) \end{aligned} \tag{32}$$

By definition of the worst-case violation, $Pr_w[AY \neq c_t|c_t = 1] \geq Pr_w[AY \neq c_m|c_m = 1]$. Moreover, when the algorithm stops, $Pr_w[c_t = 1] = Pr_w[c_m = 1]$. Therefore, the right-hand side of (32) is non-negative. Hence

$$Pr_w[AY = c_t] \geq Pr_w[AY = c_m] = 4\gamma - \rho(y) + 1 \tag{33}$$

Therefore, since $Pr_w[AY = c_t] \leq 4\gamma - \rho(y) + 1$, we conclude that $Pr_w[AY = c_t] = 4\gamma - \rho(y) + 1$ and then that

$$Pr_w[Y = 1, c_T = 1]\left(\frac{e^{\delta(c_T)}}{1 + e^{\delta(c_T)}} - \frac{1}{2}\right) = \gamma. \tag{34}$$

Small samples properties We can use a generic uniform convergence property:

Theorem 5.1. *Let $\mathbb{H}$ be a family of function mapping from $\mathfrak{X}$ to $\{-1, 1\}$ and let $S = \{x_1, ..., x_m\}$ be a sample where $x_i \sim D$ for some distribution D over $\mathfrak{X}$. With probability $1 - \delta$, for all $h \in \mathbb{H}$*

$$\left|E_{S \sim D}[h] - \frac{1}{m}\sum_{i=1}^{m} h(x_i)\right| \leq 2\mathfrak{R}_m(\mathbb{H}) + \sqrt{\frac{2\ln(1/\eta)}{m}},$$

where $\mathfrak{R}_m(\mathbb{H})$ is the Rachemacher complexity of $\mathbb{H}$.

Applying the uniform convergence result from 5.1 allows deriving small sample properties of algorithm 2. For any a sample $\{(x_i, a_i), y_i\}_{i=1}^m$ and any $c \in \mathbb{C}$ with probability $1 - \eta/2$, ,

$$\left| \sum_{i=1}^{m} w_{it} a_i y_i c_t(x_i) - E_{D_f^w} \left[\sum_{i=1}^{m} w_{it} a_i y_i c_t(x_i) \right] \right| \leq 2\mathfrak{R}_m(\mathbb{C}) + \sqrt{\frac{2\ln(2/\eta)}{m}}. \tag{35}$$

Therefore, at iteration t, with training sample $\{(x_i, a_i), y_i\}_{i=1}^m$, whenever c_t is chosen over c_o, we have with probability $1 - \eta/2$

$$Pr_w[AY = c_t | c_t] \geq 1 - \frac{2(4\gamma + 1 - 2\rho(y))}{\xi t} - 4\mathfrak{R}_m(\mathbb{C}) - \sqrt{\frac{2\ln(2/\eta)}{m}} \tag{36}$$

Therefore, with probability $1 - \eta$, the algorithm will stop with at most $\widehat{T}$ iterations where

$$\widehat{T} = \frac{2(4\gamma + 1 - 2\rho(y)}{\xi \left(\frac{1}{1+e^{\delta_m}} - 4\mathfrak{R}_m(\mathbb{C}) - \sqrt{\frac{2\ln(2/\eta)}{m}} \right)}. \tag{37}$$

Therefore, there exists a constant $\kappa_1 > 1$ such that for $m \geq \frac{2\ln(2/eta)}{(1+1/\kappa_1 - 4\mathfrak{R}(\mathbb{C}))^2}$,

$$\widehat{T} \leq T\kappa_1 \left(1 - (1 + e^{\delta_m}) \frac{4\mathfrak{R}_m(\mathbb{C}) + \sqrt{\frac{2\ln(2/\eta)}{m}}}{\xi} \right) \leq \kappa_1 T. \tag{38}$$

References

[1] Ricci et al. vs. destefano et al.

[2] How algorithms can bring down minorities credit scores? *The Atlantic*, 2016.

[3] How we analyzed the compas recidivism algorithm. *ProPublica*, 2016.

[4] Loomis vs. state of wisconsin. *Supreme Court of the State of Wisconsin*, 2016.

[5] When an algorithm helps send you to prison. *New York Times*, 2017.

[6] Caterina Calsamiglia. Decentralizing equality of opportunity. *International Economic Review*, 50(1):273–290, 2009.

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

[8] Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018.

[9] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Advances in neural information processing systems*, pages 442–450, 2010.

[10] Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. Sample selection bias correction theory. In *International Conference on Algorithmic Learning Theory*, pages 38–53. Springer, 2008.

[11] Eustasio del Barrio, Fabrice Gamboa, Paula Gordaliza, and Jean-Michel Loubes. Obtaining fairness using optimal transport theory. *arXiv preprint arXiv:1806.03195*, 2018.

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

[13] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

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

[15] Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM Journal on Computing*, 41(6):1558–1590, 2012.

[16] David A Freedman and Richard A Berk. Weighting regressions by propensity scores. *Evaluation Review*, 32(4):392–409, 2008.

[17] Arthur Gretton, Alexander J Smola, Jiayuan Huang, Marcel Schmittfull, Karsten M Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. 2009.

[18] Ursula Hébert-Johnson, Michael P Kim, Omer Reingold, and Guy N Rothblum. Calibration for the (computationally-identifiable) masses. *arXiv preprint arXiv:1711.08513*, 2017.

[19] Matthew Jagielski, Michael Kearns, Jieming Mao, Alina Oprea, Aaron Roth, Saeed Sharifi-Malvajerdi, and Jonathan Ullman. Differentially private fair learning. *arXiv preprint arXiv:1812.02696*, 2018.

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

[21] Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

[22] Michael P Kim, Omer Reingold, and Guy N Rothblum. Fairness through computationally-bounded awareness. *arXiv preprint arXiv:1803.03239*, 2018.

[23] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.