Auditing for Multi-Differential Fairness of Black Box Classifiers

January 2019

1 Multi-Differential Fairness

1.1 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.

1.2 Individual Differential Fairness

Individual Differential Fairness We define differential fairness as the guarantee that a classifier leaks with negligible probability the protected attribute of an individual.

Definition 1.1. *(Individual Differential Fairness) For $\delta \in [0, 1)$, a classifier f is $\delta-$ differential fair if for all $x \in \mathfrak{X}$ and all $a \neq a^{'} \in \mathfrak{A}$*

$$Pr[Y|a, x] \leq e^{\delta} Pr[Y|a^{'}, x] \tag{1}$$

The parameter δ controls the amount of information leaked by f on the distribution of protected attributes of an individual with auditing feature x: larger value of δ implies larger leakage. For example, for a classifier that does not satisfy the fairness condition 1.1 for some $\delta = \ln(2)$, there exist individuals x that are twice as likely to be classified $y = 1$ if $A = 1$ than if $A = -1$. In that example, the classifer's outcome y leaks information related to A that were not leaked by x alone.

Classifier's versus data's bias The advantage of individual differential fairness is to distinguish the unfairness caused by f from the one embedded already in the data. To illustrate this point, note that the fairness condition in 1.1 is equivalent to:

$$\frac{Pr[A = a|x, y]}{Pr[A = a'|x, y]} \leq e^{\delta} \frac{Pr[A = a|x]}{Pr[A = a'|x]}. \tag{2}$$

Auditing features might strongly correlate with protected attributes (e.g. zipcodes densely populated by a minority) because the data reflects social, cultural and historical biases. Individual differential fairness imposes a restriction on the classifier f to not exacerbate those biases.

Relation with differential privacy There is an analogy between individual differential fairness for classifiers and differential privacy for database queries. Differential privacy as in [2] guarantees that outcomes from a query are not distinguishable identical 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. [placeholder: why does it matter? Possibly, (i) merges the field of fairness with privacy, a field where computer science is "more comfortable" with; (ii)]

Individual Fairness The definition 1.1 is an individual level definition of fairness. Compared to the notion of individual fairness in [1], 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.

1.3 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, $\mathfrak{C}$ denotes a collection of subsets S of $\mathfrak{X}$. The collection $\mathfrak{C}_\alpha$ is α-strong if for $S \in \mathfrak{C}$ and $y \in \{-1, 1\}$, $Pr[Y = y \;\&\; x \in S] \geq \alpha$.

Definition 1.2. *(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 $S \in \mathbb{C}_\alpha$:*

$$Pr[Y = y|A = a, S] \leq e^{\delta} Pr[Y = y|A = a^{'}, S] \tag{3}$$

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 $S \in \mathfrak{C}_\alpha$. 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 3 applies only to subpopulations 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 $\delta = \infty$.

Collection of Indicators. The collection of sub-population $\mathbb{C}$ 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 [8], [6] or [5] where $\mathfrak{C}$ is the computational bound on the granularity of their definition of fairness. The richer $\mathbb{C}$, the stronger the fairness guarantee offers by definition 1.2. 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 (3) holds for *all* sub-populations of $\mathfrak{X}$, but only check the fairness condition for all sub-populations that can be *efficiently identifiable.*

2 Auditing, Agnostic Learning and PAC learning

The definition of multi-differential fairness requires to verify that in no sub-population $S \in \mathbb{C}$ with $Pr[S] \geq \alpha$, the classifier leaks information about the distribution of protected attributes. 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. In this section we show that finding an auditing algorithm reduces to agnostic learning of the class of sub-populations $\mathbb{C}$. That is, there is no $\log(\mathbb{C})$ running time auditing algorithm unless $\mathbb{C}$ is efficiently agnostically learnable.

2.1 Certifying Fairness and Agnostic Learning

Auditing for multi-differential fairness consists firstly, in establishing there exists a fairness violation; secondly, in identifying a sub-population S that violates the most the fairness condition in 1.2.

Multi Differential Fairness and Balanced Distribution The fairness condition 1.2 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 $E[u] = 1$,

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

the sub-script u indicating that the probability 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 with $Pr_w[A|x] = 1 - Pr_w[A|x]$ 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 $S \in \mathbb{C}_\alpha$

$$Pr_w[A = a|S, y] \leq \frac{e^\delta}{e^\delta + 1}, \tag{5}$$

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

$$Pr_w[A = a|S, y] - \frac{1}{2} \geq \frac{e^\delta}{e^\delta + 1} - \frac{1}{2}. \tag{6}$$

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

$$Pr_w[S, y] \left\{ Pr_w[A = a|S, y] - \frac{1}{2} \right\} \geq \gamma, \tag{7}$$

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 2.1. *(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$(\log(|\mathfrak{C}_\alpha|, \log(1/\eta), 1/\epsilon))$ samples and runs in poly$(\log(|\mathfrak{C}_\alpha|, \log(1/\eta), 1/\epsilon))$.

Searching for γ-unfairness certificate can be formulated as a problem of detecting correlations. For $g, h : \mathfrak{X} \to \{-1, 1\}$, let $\langle g, h \rangle_D \equiv E_D g(x)h(x)$ denote the average inner product between g and h. Then, S is a γ-unfair certificate of f if and only if there exists $y \in \{-1, 1\}$ such that

$$\left\langle \frac{1 + c_S}{2}, \frac{a + 1}{2} \middle| Y = y \right\rangle_{D_w^f} - \frac{1}{2} \left\langle \frac{1 + c_S}{2}, 1 \middle| Y = y \right\rangle_{D_w^f} \geq \gamma, \tag{8}$$

or equivalently, since $2Pr_w[a = c] - 1 = \langle a, c_S \rangle_{D_w^f}$

$$Pr_w[A = c_S|Y = y] \geq 1 - \rho(y) + 2\gamma \tag{9}$$

with $\rho(y) = Pr_w[A = a|Y = y]$. Detecting multi-differential fairness violation can be phrased as a weak 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*$(\log(|\mathfrak{C}, \log(\frac{1}{\delta}), \epsilon)$ outputs a function $h \in \mathbb{C}$ such that

$$\langle g, h \rangle_D + \epsilon \geq max_{c \in \mathbb{C}} \langle g, c \rangle_D.$$

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 2.1. *Let $\epsilon > 0$, $\beta > 0$ and $\mathbb{C} \subset 2^{\mathfrak{X}}$. 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 2.1 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 [3]). However, there are classes for which efficient agnostic learners exist (see [7]).

Based on theorem 2.1 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

$$opt_1 = \min_{c \in \mathbb{C}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\left(\frac{1 + c(x_i)}{2} \neq \mathbb{1}_a(a_i) \frac{1 + y_i}{2} \right) \tag{10}$$

and

$$opt_2 = \min_{c \in \mathbb{C}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\left(\frac{1 + c(x_i)}{2} \neq \mathbb{1}_a(a_i) \frac{1 - y_i}{2} \right). \tag{11}$$

The certifying algorithm 1 delivers c^*, the solution of the minimization problem corresponding to the smallest value of opt_1 and opt_2, as a $\hat{\gamma}$ unfairness certificate where $\hat{\gamma} = 1 - \min(opt_1, opt_2)$. Generic uniform convergence argument allows to derive the sample complexity and correctness of our certifying algorithm 1.

Theorem 2.2. *(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*

$$m =$$

Algorithm 1 Certifying Algorithm

```
Input: ${(x_i, f(x_i)}_{i=1}^m$, $\mathbb{C} \subset 2^{|\mathfrak{X}|}$
Find:
    $c_1^* = argmin_{c \in \mathbb{C}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\left( \frac{1 + c(x_i)}{2} \neq \mathbb{1}_a(a_i) \frac{1 + y_i}{2} \right)$ and $opt_1$
    $c_2^* = argmin_{c \in \mathbb{C}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\left( \frac{1 + c(x_i)}{2} \neq \mathbb{1}_a(a_i) \frac{1 - y_i}{2} \right)$ and $opt_2$
$\hat{\gamma} \leftarrow 1 - min\{opt_1, opt_2\}$
Return $\hat{\gamma}-$ unfair
```

2.2 Unbalanced Data

Algorithm 1 certifies efficiently multi-differential fairness. The catch is that we assume that we have oracle access to importance sampling weights w. However, most of the time, importance sampling are unobserved and need to be estimated. Moreover, variance of those estimates are known to be large (see). In this section, we propose a noise infusing technique to certify multi-differential fairness without the need to estimate importance sampling weights.

Importance Sampling and Reweighting At issue with unbalanced distribution is that the multi-differential fairness condition in (5) includes for $a \in \mathfrak{A}$ the term $w_S = Pr[A = a|S]/Pr[A \neq a|S]$:

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

Therefore, algorithm 1, when applied to unbalanced distribution, cannot distinguish the case of 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 unbalance in the data that could result from social, cultural or historical biases; the former is an issue with the classifier f itself that needs to be audited for.

One approach to obtain w is to directly estimate the density $P[A = a|x]$. This idea of importance sampling is used in propensity-score matching methods (see) in the context of counterfactual analysis. However, 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. [4] shows that the weights can be obtained by minimizing the following maximum mean discrepancy problem

$$\left\| \frac{1}{n_a} \sum_{i,A=a} w_a(x)\phi(x) - \frac{1}{n_{a'}} \sum_{i,A=a'} \phi(x) \right\|^2 \tag{13}$$

where the discrepancy is measured over all function in the RKHS represented by the kernel $k(x, x^{'}) = \langle \phi(x) | \phi(x^{'}) \rangle$. Our approach MMD takes $\phi(x) = x$ and obtains w_a by solving

$$\min_{w \in \mathbb{W}} \left\| \frac{1}{n_a} \sum_{i,A=a} w(x)x - \frac{1}{n_{a'}} \sum_{i,A=a'} x \right\|^2 + Reg(w) \tag{14}$$

where $\mathbb{W}$ is a concept class for the weight function and $Reg(w)$ is a regularization parameter for w.

The last approach is to regularize further the maximum mean discrepancy method by attempting to extract only the information predictive of the auditing algorithm outcomes ay. We jointly learn a representation ϕ of the feature space and a weight function $w(\phi)$ by minimizing the following loss:

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

In our implementation $DNN-MMD$, the common representation ϕ is learned via a neural network that is then shared with both tasks of minimizing the re-weighted auditing risk and the distributional shift between features distribution for the minority and majority group.

2.3 Fairness Diagnostic

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

Algorithm 2 Certifying Algorithm - Noise Infusion

```
Input: $\{((x_i, a_i), y_i)\}_{i=1}^m$, $\mathbb{C} \subset 2^{|\mathfrak{X}|}$, $s$, $tol$
$t = 0$; $\gamma_{-1} = 0$; $\gamma_0 = 1$
while $|\gamma_t - \gamma_{t-1}| > tol$ do
    $st$- noise infusion to transform $\{((x_i, a_i), y_i)\}_{i=1}^m$ into $\{((x_i, a_i(st)), y_i)\}_{i=1}^m$
    $c_1^* = argmin_{c \in \mathbb{C}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\left(\frac{1 + c(x_i)}{2} \neq \mathbb{1}_a(a_i(st)) \frac{1 + y_i}{2}\right)$ and $opt_1$
    $c_2^* = argmin_{c \in \mathbb{C}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\left(\frac{1 + c(x_i)}{2} \neq \mathbb{1}_a(a_i(st)) \frac{1 - y_i}{2}\right)$ and $opt_2$
    $i \leftarrow argmin_{j=1,1}\{opt_1, opt_2\}$
    $\hat{\gamma}_t \leftarrow \frac{Pr[A=a|c_i^*=1]Pr[c_i^*=1,y]}{w_{c^*==1}}$
    $t \leftarrow t + 1$
Return $\hat{\gamma}-$ unfair
```

strongest violation of multi differential fairness (i.e. with the largest value δ in 1.2). 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) create methods that ensure multi-differential fairness.

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

$$\delta_m \equiv \max_{S \in \mathbb{C}} \log \left(\frac{Pr[A = a|S, y]}{Pr[A \neq a|A, y]} \Big/ \frac{Pr[A = a|S]}{Pr[A \neq a|S]} \right). \tag{16}$$

To illustrate the challenges of the worst-violation problem, consider the case where there exists $S_0, S_\delta \in \mathbb{C}$ and a protected group $a \in \mathfrak{A}$ with no violation of multi differential fairness in S_0 (i.e. $0-$ multi differential fairness and $\delta-$ multi differential fairness violation in S_δ ($\delta > 0$) when $Y = 1$. 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 $S_\delta \cup S_0$ if the distribution is balanced. Therefore, algorithm 1 will pick indifferently c or $c^{'}$ as unfairness certificate, although $c^{'}$ does not single out S_δ as a worst violation.

Worst Violation Algorithm At issue in the previous example is that for sub-population no violation of multi differential fairness, choosing $c = 1$ or $c = -1$ will lead to same empirical risk used in 1. Algorithm 3 puts a slightly larger weight on samples $((x_i, a_i), y_i)$ whenever $a_i \neq a$ so that the empirical risk is now smaller when choosing $c = -1$ whenever there is no violation of multi differential fairness. More generally, at each iteration, the weight on samples $((x_i, a_i), y_i)$ whenever $a_i \neq a$ is increased to $1 + \nu t$, with $\nu > 0$ and the solution c_t^* of the empirical risk minimization (10) or (11) will identify $S \in \mathbb{C}$ as a violation only if $\delta \geq \log(1 + \nu t)$. The algorithm 3 terminates whenever either $|S| \leq \alpha$ or $c_t^*(x) = 1$ for all samples. At the last iteration before termination, theorem 2.3 shows that algorithm 3 will identify a sub-population with a δ-multi differential fairness violation such that

$$\delta \geq \delta_m - \tilde{O}\left(\frac{1}{\sqrt{m}}\right). \tag{17}$$

Theorem 2.3. *Suppose $\nu > 0, \epsilon > 0, \eta \in (0,1)$ and $\mathbb{C} \subset 2^{\mathfrak{X}}$ is $\alpha-$strong. Denote δ_m the worst violation of multi differential fairness for $\mathbb{C}$ as defined in* (16)*. With probability $1-\eta$, with $O()$ samples and after $O()$ iterations, algorithm 3 learns $c \in \mathbb{C}$ such that*

$$\log\left(\frac{Pr_w[A=a|y,c(x)=1]}{Pr_w[A \neq a|y,c(x)=1]}\right) \geq \delta_m - \epsilon. \tag{18}$$

Algorithm 3 Worst Violation Algorithm

```
Input: $\{((x_i,a_i),y_i)\}_{i=1}^m$, $\mathbb{C} \subset 2^{|\mathfrak{X}|}$, $s$, $tol$
$t=0$; $\gamma_{-1}=0$; $\gamma_0=1$
while $|\gamma_t - \gamma_{t-1}| > tol$ do
    $st$- noise infusion to transform $\{((x_i,a_i),y_i)\}_{i=1}^m$ into $\{((x_i,a_i(st)),y_i)\}_{i=1}^m$
    $c_1^* = argmin_{c\in\mathbb{C}} \frac{1}{m}\sum_{i=1}^m \mathbb{1}\left(\frac{1+c(x_i)}{2} \neq \mathbb{1}_a(a_i(st))\frac{1+y_i}{2}\right)$ and $opt_1$
    $c_2^* = argmin_{c\in\mathbb{C}} \frac{1}{m}\sum_{i=1}^m \mathbb{1}\left(\frac{1+c(x_i)}{2} \neq \mathbb{1}_a(a_i(st))\frac{1-y_i}{2}\right)$ and $opt_2$
    $i \leftarrow argmin_{j=1,1}\{opt_1, opt_2\}$
    $\hat{\gamma}_t \leftarrow \frac{Pr[A=a|c_i^*=1]Pr[c_i^*=1,y]}{w_{c^*==1}}$
    $t \leftarrow t+1$
Return $\hat{\gamma}-$ unfair
```

3 Experiments

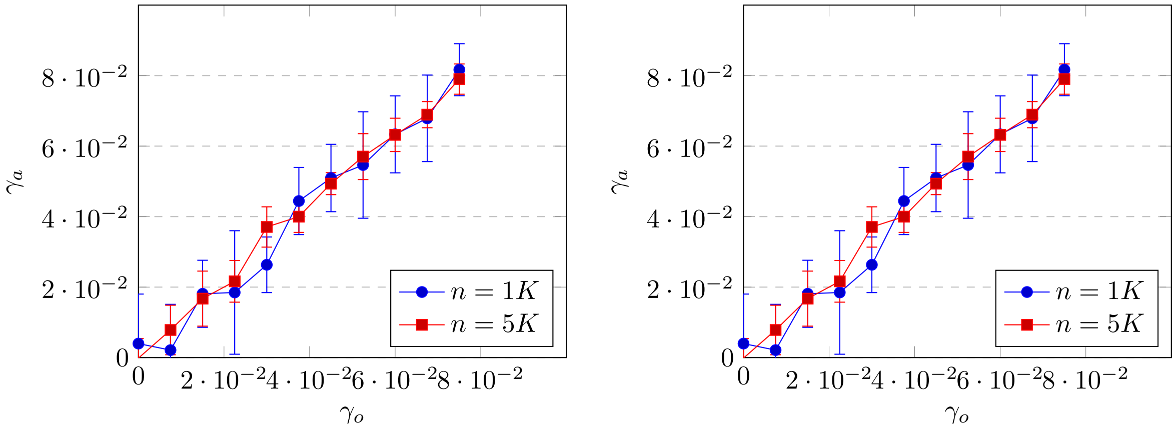
3.1 Synthetic Data

A synthetic data is constructed by drawing independently two features X_1 and X_2 from two normal distribution $N(A\mu, 1)$, where $\mathfrak{A} = \{-1,1\}$ is the protected attribute and $\mu \geq 0$ is an unbalance factor. $\mu = 0$ means that the data is perfectly balanced. As μ increases, the distriubtion $Pr(x|A=1)$ becomes more dense in the right uppermost quadrant than the distribution $Pr[x|A=-1]$. The data is labeled according to the sign of $X_1 + X_2 + e$, 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 metric-free individual unfairness: with probability $1-\nu \in [0,1)$, the sign of the outcomes from individuals with $A=-1$ is changed from -1 to $+1$ if $x_1^2 + x_2^2 \leq 1$ and $x_1 + x_2 \leq 0$; fi $A=1$, all classifier's outcomes are set to 1 if if $x_1^2 + x_2^2 \leq 1$ and $x_1 + x_2 \leq 0$. For $\nu = 0$, the audited classifier is metric-free individual fair; however, as ν increases, the half circle $\{(x_1,x_2)|x_1^2 + x_2^2 \leq 1 \text{ and } \}$ 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 First, the auditing algorithm *AMDF* is trained using a short decision tree (with a maximm depth of 5) and a balanced data ($\mu = 0$). 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 algorithm *AMDF*. Figure **??** plots the average ratio γ_a/γ_o after 100 simulations for value of ν varying from 0 to 1. The experiment is conducted with data set of size

$n \in \{5000, 10000, 20000, 30000\}$. As the classifier's unfairness increases, the auditor bias decreases since the ratio γ_a/γ_o converges to one. This result is consistent with the correctness of algorithm 1 proved in theorem 2.2. As the sample size increases, the bias at low level of unfairness is reduced.

Concept Class Complexity Figure **??** runs a similar experiment 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 2.2 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 $\mathbb{C}$.

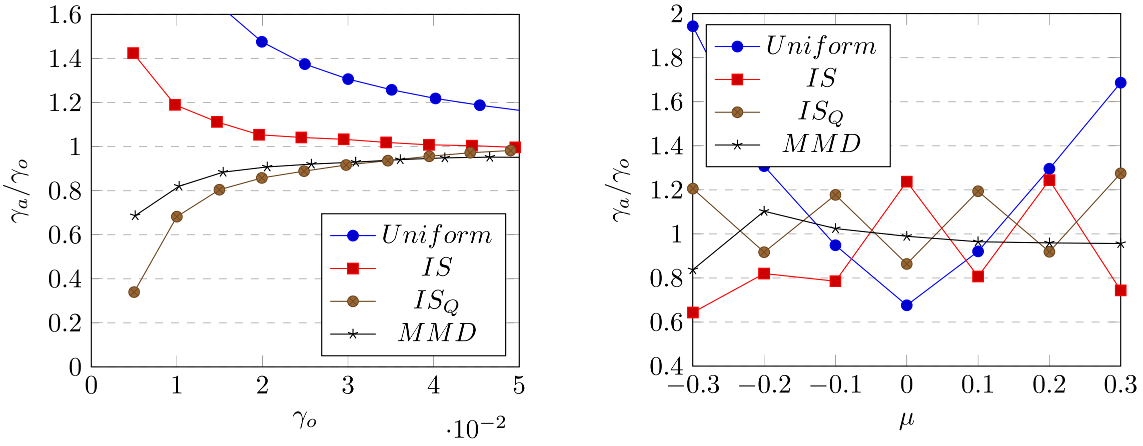


(a) Effect of unfairness intensity on auditor's performance.

(b) Effect of sub-population size α on auditor's performance.

Figure 1: Certifying γ- multi differential unfairness. Figure **??**: the auditor is a decision tree with depth 5; sample size varies is $10K$ and the unbalance parameter, μ is set to 0.25; Figure **??**: the auditor is a decision tree with depth varying from 5 to 40; sample size is $10K$. The auditor's bias is measured by the ratio γ_a/γ_o: a ratio of 1 means that the auditor measures an unbiased level of unfairness.

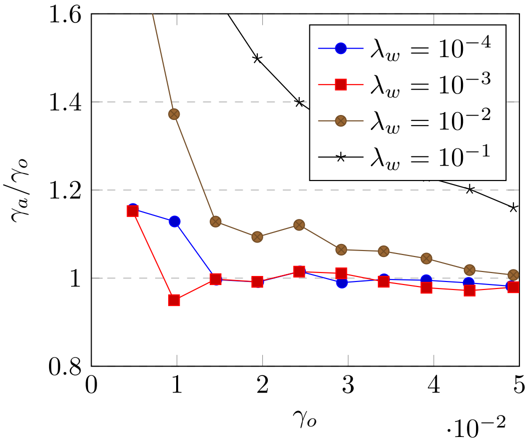
Unbalanced Data To test the performance of our certifying algorithm on unbalanced data, we repeat the previous experiment with $\mu = 0.25$. We compare the performance of four reweighting approaches: uniform weight (UW); estimated importance sampling weights (IPW); quantile weights (QW) with 10 bins. Figure **??** shows that using absent of a reweighting scheme (UW) the certifying algorithm fails to estimate correctly the classifier's unfairness: for low level of unfairness ($\gamma_o < 2.10^{-2}$), the bias in γ_a is over 100%.



(a) Varying classifier's unfairness γ_o.

(b) Varying data unbalance μ.

Figure 2: Certifying γ- multi differential unfairness when data is unbalanced. Figure **??**: the auditor is a decision tree with depth 5; sample size is $10K$; Figure **??**: the auditor is a decision tree with depth 5; sample size is $10K$; $\gamma_0 = 3$. The auditor's bias is measured by the ratio γ_a/γ_o: a ratio of 1 means that the auditor measures an unbiased level of unfairness.



(a) Varying regularization λ_w.

Figure 3: Certifying γ- multi differential unfairness when data is unbalanced. The auditor is a decision tree with depth 5; sample size is $10K$; The auditor's bias is measured by the ratio γ_a/γ_o: a ratio of 1 means that the auditor measures an unbiased level of unfairness.

4 Appendix

4.1 Analysis of Algorithm 2

Let $a \in \mathfrak{A}$. Without loss of generality we consider the case where $Y = 1$. At each iteration t, denote c_t^* the solution of the following optimization problem

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

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.

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

For $c \in \mathbb{C}$, denote $\beta_c = E_{c=1,y=1}[\mathbb{1}_a(a_i)]$. Let $c_0 \in \mathbb{C}$ such that $c_0(x) = -1$ for all $x \in \mathfrak{X}$. Denote $B_c = \{x_i | g(x_i) = 1 \; f_a(x_i) = f_{a'}(x_i)\}$, $B_g^+ = \{x_i | g(x_i) = 1 \; f_a(x_i) > f_{a'}(x_i)\}$ and $B_g^- = \{x_i | g(x_i) = 1 \; f_a(x_i) < f_{a'}(x_i)\}$. Observe that

$$\beta_g = \left| \sum_{i, x_i \in B_g^+} w_i - \sum_{i, x_i \in B_g^-} w_i \right|. \tag{21}$$

Assume first that $\sum_{i, x_i \in B_g^+} w_i > \sum_{i, x_i \in B_g^-} w_i$. Therefore,

$$\begin{aligned} E_D \left[\sum_{i=1, g(x_i)=1}^{m} w_{it} f_i^* g(x_i) \right] &= E_D \left[\sum_{i=1, x_i \in B_g}^{m} w_{it} f_i^* + \sum_{i=1, x_i \notin B_g}^{m} w_{it} f_i^* \right] \\ &= -\frac{1}{2} \epsilon t E_D[|B_g|] - \epsilon t E_D[|B_g^-|] - E_D \left[\sum_{i=1, x_i \in B_g^-}^{m} w_i \right] \\ &+ E_D \left[\sum_{i=1, x_i \in B_g^+}^{m} w_i \right] \\ &= -\frac{1}{2} \epsilon t E_D[|B_g|] - \epsilon t E_D[|B_g^-|] + \beta_g, \end{aligned} \tag{22}$$

where $|B| = \sum_{i=1, x_i \in B}^{m} w_i$ for any $B \in \mathbb{C}$. Moreover,

$$E_D \left[\frac{1}{m} \sum_{i=1, g(x_i)=1}^{m} w_{it} f_i^* g_0(x_i) \right] = \frac{1}{2} \epsilon t E_D[|B_g|] + \epsilon t E_D[|B_g^-|] - \beta_g. \tag{23}$$

Therefore, g cannot be a solution of (**??**) if

$$\epsilon t \left(E_D[|B_g|] + 2 E_D[|B_g^-|] \right) > 2\beta_g. \tag{24}$$

Note that $|B_g| = 1 - \left(|B_g^-| + |B_g^+| \right)$ and $\beta_g = |B_g^+| - |B_g^-|$. Therefore, g cannot be a solution of (**??**) if

$$t > \frac{2\beta_g}{\epsilon(1-\beta_g)} \tag{25}$$

At any iteration t, a solution of (**??**) is either $g(x) = -1$ for all $x \in \mathfrak{X}$ or $\beta_g > \frac{\epsilon t}{\epsilon t + 2}$.

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

Theorem 4.1. *Let$\mathfrak{H}$ 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 $h \in \mathfrak{H}$*

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

Applying the uniform convergence result from **??** allows deriving property of algorithm 2. For any a sample S and any $g \in \mathbb{C}$ with probability $1-\delta/2$, ,

$$\left| \frac{1}{m} \sum_{i=1, g(x_i)=1}^{m} w_{it} f_i^* g(x_i) + \frac{1}{2}\epsilon t E_D[|B_g|] + \epsilon t E_D[|B_g^-|] - \beta_g \right| \leq 2\mathfrak{R}_m(\mathbb{C}) + \sqrt{\frac{2\ln(2/\delta)}{m}}, \tag{26}$$

and

$$\left| \frac{1}{m} \sum_{i=1, g(x_i)=1}^{m} w_{it} f_i^* g_0(x_i) - \frac{1}{2}\epsilon t E_D[|B_g|] - \epsilon t E_D[|B_g^-|] + \beta_g \right| \leq 2\mathfrak{R}_m(\mathbb{C}) + \sqrt{\frac{2\ln(2/\delta)}{m}}. \tag{27}$$

Therefore, with probability $1-\delta$, h cannot be solution of the empirical counterpart of (**??**) if

$$t > \frac{2\beta_g}{\epsilon(1-\beta_g)} + \frac{4}{1-\beta_g}\mathfrak{R}_m(\mathbb{C}) + \frac{2}{1-\beta_g}\sqrt{\frac{2\ln(2/\delta)}{m}}. \tag{28}$$

Therefore at iteration t, with probability $1-\delta$, a solution of (**??**) for a sample S is either $g(x) = -1$ for all $x \in S$ or

$$\beta_g > \frac{\epsilon t}{2+\epsilon t} - \frac{4}{2+\epsilon t}\mathfrak{R}_m(\mathbb{C}) - \frac{2}{2+\epsilon t}\sqrt{\frac{2\ln(2/\delta)}{m}}. \tag{29}$$

References

[1] 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.

[2] 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.

[3] 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.

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

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

[6] 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.

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

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