Fairness Checking

1 Introduction

[Deb: Rough sketch of the introduction:

- Usual Motivation for fairness
- Why do we care about distributional robustness in this setting?
- Connection with verification literature
- Related work – (a) fairness in classification, (b) distributionally robust optimization,
- Contributions – (a) distributional robustness for fairness, (b) general framework to design such a classifier, (c) faster approximate minmax solution, (d) simulation results

] Nowadays, AI systems are increasingly used in various high-stakes decision making scenarios. Applications include bail decision, credit approval, and housing allocation, to name a few. Often these applications use learning algorithms trained on past biased data, and such bias is often reflected in the eventual decision. For example, [3] show that popular word embeddings implicitly encode societal biases, such as gender norms. Similarly, [4] evaluate existing facial recognition systems and find that they perform better on lighter-skinned subjects as a whole than on darker-skinned subjects as a whole with an 11.8% - 19.2% difference in error rates. To mitigate these biases, there have been several approaches in the ML fairness community to design fair classifiers [11, 8, 2].

However, the literature has largely ignored the robustness of such fair classifiers. As an example, we consider the performance of the optimized pre-processing algorithm [5] on the popular COMPAS dataset [1]. As a metric of fairness we consider the notion of *demographic parity* (DP), which measures the difference in accuracy between two protected groups. Figure shows two situations – (1) unweighted training distribution (in blue), and (2) weighted training distributions. The optimized pre-processing algorithm [5] is almost fair on the unweighted training dataset (DP ≤ 0.02). However, it shows demographic parity of at least 0.1 on the weighted dataset, despite the fact that the marginal distributions of the features look almost the same for the two scenarios. This example motivates our work and we aim to design a fair classifier that is robust to such perturbations. We also show how to construct such reweighted examples.

Nonetheless, since different algorithms adopt different definitions of fairness and provide different trade-offs with respect to accuracy and utility, it is neither legal nor ethical to enforce businesses to use such algorithms. In this paper, we approach this problem with a perspective from the literature of automated verification, and aim to build tools that can verify whether an algorithm satisfies a given fairness criteria irrespective of the particular algorithm or dataset used. We show using these tools that, although current group fairness algorithms may mitigate fairness for a specific distribution of data, slight perturbations to that data's distribution result in violations of the fairness criteria.

2 Problem and Definitions

Let $\mathcal{W}$ be the set of all possible weights i.e. $\mathcal{W} = \{w \in \mathbb{R}^+_n : \sum_i w_i = 1\}$. For a hypothesis h and weight w, we define the following loss function $\ell(h, w) = \sum_{i=1}^n w_i \ell(h(x_i, a_i), y_i)$, where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a convex loss function. Note that, this does not pose any restriction on the classifier h, which can be any arbitrary classifier like neural network. We also use $\delta^w_F(f)$ to define

Preprint. Under review.

the "unfairness gap" with respect to the weighted empirical distribution defined by the weight w and fairness constraint F (e.g. DP, EOY1, EOY0). For example, $\delta^w_{DP}(f)$ is defined as

$$\delta^w_{DP}(f) = \left| \frac{\sum_{i:a_i=a} w_i f(x_i, a)}{\sum_{i:a_i=a} w_i} - \frac{\sum_{i:a_i=a'} w_i f(x_i, a')}{\sum_{i:a_i=a'} w_i} \right|.$$

For the remainder of this section, we will work with demographic parity (DP), but other types of fairness constraints can be handled analgously. For a class of hypothesis $\mathcal{H}$, let $\mathcal{H}_{\mathcal{W}} = \{h \in \mathcal{H} : \delta^w_F(h) \leq \epsilon \; \forall w \in \mathcal{W}\}$ be the set of feasible hypothesis. Our goal is to solve the following minmax problem:

$$\min_{h \in \mathcal{H}_{\mathcal{W}}} \max_{w \in \mathcal{W}} \ell(h, w) \tag{1}$$

We will allow our algorithm to output a classifier which is randomized i.e. it is a distribution over the hypothesis $\mathcal{H}$. This will also be necessary if the space $\mathcal{H}$ is non-convex or if the fairness constraints are such that the set of feasible hypothesis $\mathcal{H}_{\mathcal{W}}$ is non-convex. Let us write $\Delta(\mathcal{H}_{\mathcal{W}})$ to denote a distribution over the space of feasible hypothesis. For a randomized classifier $Q \in \mathcal{H}_{\mathcal{W}}$ define the expected loss of Q as $\ell(Q, w) = \sum_h Q(h)\ell(h, w)$.

3 Meta Algorithm

In this section, we first provide a meta-algorithm that helps us to design fair classifiers that are robust with respect to any distribution that are some weighted perturbations of the empirical distribution of the training data. The meta-algorithm repeatedly calls an oracle that solves the fair classification problem with respect to a given weighted empirical distribution. In the next section, we will see how to design such an oracle by modifying standard fair classifiers.

ALGORITHM 1: Meta-Algorithm

Input: Training Set: $\{x_i, a_i, y_i\}_{i=1}^n$, set of weights: $\mathcal{W}$, hypothesis class $\mathcal{H}$, parameters T and η.
$w_0(i) = 1/n$ for all $i \in [n]$
$h_0 \in \arg\min_{h \in \mathcal{H}_{\mathcal{W}}} \sum_{i=1}^n w_0(i)\ell(h(x_i, a_i), y_i)$
for *each time step* $t \in [T]$ **do**
  $w_t = w_{t-1} + \eta \nabla_w \ell(h_{t-1}, w_{t-1})$
  $w_t = \Pi_{\mathcal{W}}(w_t)$
  $h_t = M(w_t)$ [Approximate solution of $\min_{h \in \mathcal{H}_{\mathcal{W}}} \sum_{i=1}^n w_t(i)\ell(h(x_i, a_i), y_i)$]
end
Output: Uniform distribution over $\{h_1, \ldots, h_T\}$.

Algorithm 1 provides a meta algorithm to solve the min-max optimization problem defined in equation 1. The algorithm is based on ideas presented in [6], which, given an α-approximate Bayesian oracle for distributions over loss functions, provides an α-approximate robust solution. So we assume an access to the following approximate Bayesian oracle.

Definition 1. *For any weight $w \in \mathbb{R}^n_+$, an α-approximate oracle M returns a hypothesis $h' = M(w)$ such that*

$$\sum_{i=1}^n w_i \ell(h'(x_i, a_i), y_i) \leq \alpha \min_{h \in \mathcal{H}_{\mathcal{W}}} \sum_{i=1}^n w_i \ell(h(x_i, a_i), y_i).$$

Using the approximate Bayesian oracle, we have the following gurantee on the output of algorithm 1.

Theorem 1. *Suppose the loss function $\ell(\cdot, \cdot)$ is convex in its first argument. Then the ensemble hypothesis $h^* = \frac{1}{T}\sum_{t=1}^T h_t$, where $\{h_1, \ldots, h_T\}$ are output by the meta-algorithm 1 given access to the α-approximate oracle (1), satisfies the following:*

$$\max_{w \in \mathcal{W}} \mathbb{E}_{h \sim h^*} \left[\sum_{i=1}^n w_i \ell(h(x_i, a_i), y_i) \right] \leq \alpha \min_{h \in \mathcal{H}_{\mathcal{W}}} \max_{w \in \mathcal{W}} \ell(h, w) + \max_{w \in \mathcal{W}} \|w\|_2 \sqrt{\frac{2}{T}}$$

Proof. Use theorem 7 from **(author?)** [6]. □

We now derive an algorithm for the Baysian oracle promised in 1. We first discretize the set of weights $\mathcal{W}$. For each $i \in [n]$, consider the buckets $B_0 = [0, \delta)$, $B_{j+1} = [(1+\gamma_1)^j\delta, (1+\gamma_1)^{j+1}\delta)$ for $j = 0, 1, \ldots, M-1$ for $M = O(\log_{1+\gamma_1}(1/\delta))$. For any weight $w \in \mathcal{W}$, we consider the weight w'. Here w'_i is the upper-end point of the bucket containing w_i. Note that this guarantees that either $w_i \leq \delta$ or $\frac{w'_i}{1+\gamma_1} \leq w_i \leq w'_i$. Now we show that fairness guarantee with respect to the weight w' is sufficient to guarantee fairness with respect to the weight w.

$$\frac{\sum_{i:a_i=a} w_i f(x_i, a)}{\sum_{i:a_i=a} w_i} \geq \frac{1}{1+\gamma_1} \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i} \geq (1-\gamma_1) \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i}$$

Also note that,

$$\begin{aligned}\sum_{i:a_i=a} w'_i &\leq \sum_{i:a_i=a, w_i>\delta} w_i + \sum_{i:a_i=a, w_i\leq\delta} \delta \\ &\leq (1+\gamma_1) \sum_{i:a_i=a, w_i>\delta} w'_i + n\delta\end{aligned}$$

This gives us the following.

$$\frac{\sum_{i:a_i=a} w_i f(x_i, a)}{\sum_{i:a_i=a} w_i} \leq \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\frac{1}{1+\gamma_1}\sum_{i:a_i=a} w'_i - \frac{n\delta}{1+\gamma_1}} \leq (1+\gamma_1) \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i - n\delta}$$

Now we substitute, $\delta = \gamma_1/(2n)$.

$$\frac{\sum_{i:a_i=a} w_i f(x_i, a)}{\sum_{i:a_i=a} w_i} \leq (1+\gamma_1) \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i - \gamma_1/2} \leq \frac{1+\gamma_1}{1-\gamma_1} \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i} \leq (1+3\gamma_1) \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i} \tag{2}$$

Now we bound $\delta^w_{DP}(f)$ using the results above. Suppose

$$\delta^w_{DP}(f) = \frac{\sum_{i:a_i=a} w_i f(x_i, a)}{\sum_{i:a_i=a} w_i} - \frac{\sum_{i:a_i=a'} w_i f(x_i, a')}{\sum_{i:a_i=a'} w_i}$$

Then we have,

$$\begin{aligned}\delta^w_{DP}(f) &\leq (1+3\gamma_1) \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i} - (1-\gamma_1) \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i} \\ &\leq \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i} - \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i} + 4\gamma_1 \\ &\leq \delta^{w'}_{DP}(f) + 4\gamma_1\end{aligned}$$

Therefore, if we guarantee that $\delta^{w'}_{DP}(f) \leq \varepsilon - 4\gamma_1$, we have $\delta^w_{DP}(f) \leq \varepsilon$. Therefore, in order to ensure that $\delta^w_{DP}(f) \leq \varepsilon$ we construct $M = O(\log_{1+\gamma_1}(2n/\gamma_1))$ buckets and enforce $\varepsilon - 4\gamma_1$ fairness for all the weights constructed using the end-points of the bucket. Let us write $N(\gamma_1, \mathcal{W})$ to denote the set of all possible such weights vectors. We also introduce the notation $T(w, a, f) = \frac{\sum_{i:a_i=a} w_i f(x_i, a)}{\sum_{i:a_i=a} w_i}$. Then $\delta^w_{DP}(f) = \sup_{a,a'} |T(w, a, f) - T(w, a', f)|$.

4 Approximate Fair Classifier

Now our aim is to solve the following problem.

$$\begin{aligned}&\min_{h\in\mathcal{H}} \sum_{i=1}^n w^0_i \ell(h(x_i, a_i), y_i) \\ &\text{s.t.} T(w, a, h) - T(w, a', h) \leq \varepsilon - 4\gamma_1 \; \forall w \in N(\gamma_1, \mathcal{W}) \; \forall a, a' \in \mathcal{A}\end{aligned} \tag{3}$$

We form the following Lagrangian.

$$\min_{h\in\mathcal{H}} \max_{\substack{\lambda\in\mathbb{R}_+^{|N(\gamma_1,\mathcal{W})|\times|\mathcal{A}|^2} \\ \|\lambda\|_1\le B}} \sum_{i=1}^{n} w_i^0 \ell(h(x_i,a_i),y_i) + \sum_{w\in N(\gamma_1,\mathcal{W})} \sum_{a,a'\in\mathcal{A}} \lambda_w^{a,a'} (T(w,a,h) - T(w,a',h) - \varepsilon + 4\gamma_1) \tag{4}$$

We now focus on solving the problem defined in equation 4. In order to do so, we first convert equation 4 as a two-player zero-sum game. Here the learner's pure strategy is to play a hypothesis h in $\mathcal{H}$. And the auditor's pure strategy is to play a vector $\lambda \in \mathbb{R}_+^{|N(\gamma_1,\mathcal{W})|\times|\mathcal{A}|^2}$ such that either all the coordinates of λ are zero or exactly one is set to B. We denote these set of pure strategies by Λ_p. Then for any pair of actions $(h,\lambda) \in \mathcal{H}\times\Lambda_p$, the payoff is defined as

$$U(h,\lambda) = \sum_{i=1}^{n} w_i^0 \ell(h(x_i,a_i),y_i) + \sum_{w\in N(\gamma_1,\mathcal{W})} \sum_{a,a'\in\mathcal{A}} \lambda_w^{a,a'} (T(w,a,h) - T(w,a',h) - \varepsilon + 4\gamma_1)$$

Now our goal is to compute a ν-approximate minmax equilibrium of this game. First, we see how both the h-player and the λ-player compute their best responses.

Best response of the h-player: For each $i \in [n]$, we introduce the following notation

$$\Delta_i = \sum_{w\in N(\gamma_1,\mathcal{W})} \sum_{a'\neq a_i} \left(\lambda_w^{a_i,a'} - \lambda_w^{a',a_i}\right) \frac{w_i}{\sum_{j:a_j=a_i} w_j}$$

With this notation, the payoff becomes

$$U(h,\lambda) = \sum_{i=1}^{n} w_i^0 \ell(h(x_i,a_i),y_i) + \Delta_i h(x_i,a_i) - (\varepsilon - 4\gamma_1) \sum_{w\in N(\varepsilon/5,\mathcal{W})} \sum_{a,a'\in\mathcal{A}} \lambda_w^{a,a'}$$

Let us introduce the following costs.

$$c_i^0 = \begin{cases} \ell(0,1)w_i^0 & \text{if } y_i = 1 \\ \ell(0,0)w_i^0 & \text{if } y_i = 0 \end{cases} \qquad c_i^1 = \begin{cases} \ell(1,1)w_i^0 + \Delta_i & \text{if } y_i = 1 \\ \ell(1,0)w_i^0 + \Delta_i & \text{if } y_i = 0 \end{cases} \tag{5}$$

Then the h-player's best response becomes the following cost-sensitive classification problem.

$$\widehat{h} \in \arg\min_{h\in\mathcal{H}} \sum_{i=1}^{n} \left\{c_i^1 h(x_i,a_i) + c_i^0 (1 - h(x_i,a_i))\right\} \tag{6}$$

Therefore, as long as we have access to an oracle that solves the cost-sensitive classification problem, the h-player can solve it's best response problem.

Best response of the λ-player: We first discretize the simplex over $|\mathcal{A}|$ groups, $\Delta_{\mathcal{A}} = \{\pi \in \mathbb{R}_+^{|\mathcal{A}|} : \sum_{a\in\mathcal{A}} \pi_a = 1\}$. First, discretize $[0,1]$ as $0, \delta, (1+\gamma_2)^j\delta$ for $j = 1,2,\ldots,M$ for $M = O(\log_{1+\gamma_2}(1/\delta))$. This discretizes $[0,1]^{\mathcal{A}}$ into $M^{|\mathcal{A}|}$ points. Now we just retain the points for which $\sum_{a\in\mathcal{A}} \pi_a \in (1-2\gamma_2, 1+2\gamma_2)$ and discard all other points. Let us denote the set of such points as $N(\gamma_2,\mathcal{A})$. Algorithm 2 describes the best response of the λ-player for a given choice of h. It goes through all the points π in $N(\gamma_2,\mathcal{A})$ and for each such value and a pair of groups a,a' finds the weight w which maximizes $T(w,a,h) - T(w,a',h)$. Note that this can be solved using a Linear Program as the weights assigned to a group is fixed by the point π. Out of all the solutions, the algorithm finds the one with the maximum value. Then it checks whether the maximum violates the constraint i.e. greater than $\varepsilon - 4\gamma_1$. If so, it sets the corresponding λ value to B and everything else to 0. If not, it returns the zero vector. Note that, the weight returned by the linear program need not

correspond to a weight in $N(\gamma_1, \mathcal{W})$. In that case, the algorithm rounds the weight to the nearest weight in $N(\gamma_1, \mathcal{W})$ and sets the corresponding λ variable.

ALGORITHM 2: Best Response of the λ-player

Input: Training Set: $\{x_i, a_i, y_i\}_{i=1}^n$, and hypothesis $h \in \mathcal{H}$.
for *each* $\pi \in N(\gamma_2, \mathcal{A})$ **do**
  for *each* $a, a' \in \mathcal{A}$ **do**
    Solve the following LP:

$$
\begin{aligned}
w(a, a', \pi) = \arg\max_{w} \quad & \frac{1}{\pi_a} \sum_{i:a_i=a} w_i h(x_i, a) - \frac{1}{\pi_{a'}} \sum_{i:a_i=a'} w_i h(x_i, a') \\
\text{s.t.} \quad & \sum_{i=1:a_i=a} w_i = \pi_a \\
& \sum_{i=1:a_i=a'} w_i = \pi_{a'} \\
& w_i \geq 0 \quad \forall i \in [n] \\
& \sum_{i=1}^{n} w_i = 1
\end{aligned}
$$

    Set $\text{val}(a, a', \pi) = \frac{1}{\pi_a} \sum_{i:a_i=a} w(a, a', \pi)_i h(x_i, a) - \frac{1}{\pi_{a'}} \sum_{i:a_i=a'} w(a, a', \pi)_i h(x_i, a')$
  end
end
Set $(a^*, a'^*, \pi^*) = \arg\max_{a,a',\pi} \text{val}(a, a', \pi)$
if *val*$(a^*, a'^*, \pi^*) > \varepsilon$ **then**
  Let $w = w(a^*, a'^*, \pi^*)$.
  for $i \in [n]$ **do**
    Let w_i' be the upper-end point of the bucket containing w_i.
  end
  return $\lambda_w^{a,a'} = \begin{cases} B & \text{if } (a, a', w) = (a^*, a'^*, w') \\ 0 & \text{o.w.} \end{cases}$
end
else
  return $\lambda_w^{a,a'} = 0$ *for all* $a, a' \in \mathcal{A}$ *and* $w \in N(\gamma_1, \mathcal{W})$.
end

Theorem 2. *Algorithm 2 is an $B(4\gamma_1 + \gamma_2)$-approximate best response for the λ-player i.e. for any $h \in \mathcal{H}$, it returns λ^* such that*

$$U(h, \lambda^*) \geq \max_{\lambda} U(h, \lambda) - B(4\gamma_1 + \gamma_2)$$

Proof. We need to consider two cases. First, suppose that $T(w, a, h) - T(w, a', h) \leq \varepsilon - 4\gamma_1$ for all $w \in N(\gamma_1, \mathcal{W})$ and $a, a' \in \mathcal{A}$. Then for any marginal $\pi \in N(\gamma_2, \mathcal{A})$, and a, a' consider the corresponding linear program. We show that the optimal value of the LP is bounded by ε. Indeed, any weight w satisfying the marginal conditions i.e. $\sum_{i:a_i=a} w_i = \pi_a$ and $\sum_{i:a_i=a'} w_i = \pi_{a'}$. Then w' be the weight constructed by rounding the weight w i.e. for each $i \in [n]$, let w_i' be the upper-end point of the bucket containing w_i. As we proved earlier $\delta_{DP}^{w}(h) \leq \delta_{DP}^{w'} + 4\gamma_1$. This gives that $\delta_{DP}^{w}(h) \leq \varepsilon$. This implies that the optimal value of the LP is always less than ε. So algorithm 2 returns the zero vector, which is the optimal solution in this case.

Second, there exists w, a, a' such that $T(w, a, h) - T(w, a', h) > \varepsilon - 4\gamma_1$ and in particular let $(w^*, a^*, a'^*) \in \arg\max_{w,a,a'} T(w, a, h) - T(w, a', h)$. Then the optimal solution sets $\lambda_{w^*}^{a^*,a'^*}$ to B and everything else to zero. Let π_{a^*} and $\pi_{a'^*}$ be the corresponding marginals for groups a and a', and let π'_{a^*} and $\pi'_{a'^*}$ be the upper-end point of the bucket containing π_{a^*} and $\pi_{a'^*}$ respectively. This guarantees the following.

$$\frac{\pi'_{a^*}}{1 + \gamma_2} \leq \pi_{a^*} \leq \pi'_{a^*} \quad \text{and} \quad \frac{\pi'_{a'^*}}{1 + \gamma_2} \leq \pi_{a'^*} \leq \pi'_{a'^*}$$

Now, consider the LP corresponding to the marginal π' and subgroups a^* and a'^*.

$$\begin{aligned}
&\frac{1}{\pi'_{a^*}} \sum_{i:a_i=a^*} w_i h(x_i, a^*) - \frac{1}{\pi'_{a'^*}} \sum_{i:a_i=a'^*} w_i h(x_i, a'^*) \\
&\geq \frac{1}{(1+\gamma_2)\pi_{a^*}} \sum_{i:a_i=a^*} w_i h(x_i, a^*) - \frac{1}{\pi_{a'^*}} \sum_{i:a_i=a'^*} w_i h(x_i, a'^*) \\
&\geq (1-\gamma_2) T(w, a^*, h) - T(w, a'^*, h) \\
&\geq T(w, a^*, h) - T(w, a'^*, h) - \gamma_2
\end{aligned}$$

Therefore, if the maximum value of $T(w, a, h) - T(w, a', h)$ over all weights w and subgroups a, a' is larger than $\varepsilon + \gamma_2$, the value of the corresponding LP will be larger than ε and the algorithm will set the correct coordinate of λ to B. On the other hand, if the maximum value of $T(w, a, h) - T(w, a', h)$ is between $\varepsilon - 4\gamma_1$ and $\varepsilon + \gamma_2$. In that case, the algorithm might return the zero vector with value zero. However, the optimal can be as large as $B \times (4\gamma_1 + \gamma_2)$. □

We are now ready to introduce our algorithm for the problem defined in equation 4. In this algorithm, the h-player will use a learning algorithm, but the λ-player will use algorithm 2 to compute approximate best response. We first recall Regularized Follow the Leader (RFTL) algorithm and its guarantees (c.f. [9]).

ALGORITHM 3: RFTL

Input: $\eta > 0$, regularization function R, and a convex compact set $\mathcal{K}$.
Set $x_1 = \arg\min_{x \in \mathcal{K}} R(x)$
for $t \in [T]$ **do**
  Predict x_t
  Observe f_t and compute $\nabla f_t(x_t)$
  Update

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \left\{ \eta \sum_{s=1}^{t} \nabla f_t(x_t)^T x + R(x) \right\}$$

end

Theorem 3. *The RFTL algorithm achieves the following regret bound for any $u \in \mathcal{K}$*

$$\sum_{t=1}^{T} f_t(x_t) - f_t(u) \leq \frac{\eta}{4} \sum_{t=1}^{T} \|\nabla f_t(x_t)\|_\infty^2 + \frac{R(u) - R(x_1)}{2\eta}$$

Moreover, if $\|\nabla f_t(x_t)\|_\infty \leq G_R$ for all t and $R(u) - R(x_1) \leq D_R$ for all $u \in \mathcal{K}$, then we can optimize η to get the following bound: $\sum_{t=1}^{T} f_t(x_t) - f_t(u) \leq D_R G_R \sqrt{T}$.

Recall the best response of the h-player. For a given λ the best response of the h-player is the following cost-sensitive classification problem.

$$\widehat{h} \in \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} c_i^1(\lambda) h(x_i, a_i) + c_i^0(\lambda)(1 - h(x_i, a_i)) \tag{7}$$

Writing $L_i(\lambda) = c_i^1(\lambda) - c_i^0(\lambda)$ the problem stated above becomes

$$\widehat{h} \in \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} L_i(\lambda) h(x_i, a_i) \tag{8}$$

Algorithm 4 describes the algorithm for solving a minmax approximate equilibrium of the game $U(h, \lambda)$ for $h \in \mathcal{H}$ and $\lambda \in \mathbb{R}_+^{|N(\gamma_1, \mathcal{W})| \times |\mathcal{A}|^2}$, $\|\lambda\|_1 \leq B$. We will later see how this solution immediately leads to a solution for the optimization problem defined in equation 3. The h-player uses the RFTL algorithm as a learning algorithm whereas the λ-player approximately best respond to h_t in each round. Recall that, in order to use the RFTL algorithm we need to specify the regularization function R and cost function f_t in each round. We choose $R(x) = 1/2\|x\|_2^2$. As the learner always chooses a vector in $\{0, 1\}^n$ corresponding to the n predictions for the n training instances, the

diameter D_R is bounded by n. At round t, for an action h_t, the cost function is $f_t(h_t) = U(h_t, \lambda_t)$ where λ_t is the $B(4\gamma_1 + \gamma_2)$-approximate best-response to h_t. Now we show that the optimization problem faced by the learner becomes a cost-sensitive classification problem. Indeed,

$$\begin{aligned}
&\eta \sum_{s=1}^{t} \langle L(\lambda_s), h \rangle + R(h) \\
&= \eta \sum_{s=1}^{t} \sum_{i=1}^{n} L(\lambda_s) h(x_i, a_i) + \frac{1}{2} \sum_{i=1}^{n} (h(x_i, a_i))^2 \\
&= \eta \sum_{i=1}^{n} L(\sum_{s=1}^{t} \lambda_s) h(x_i, a_i) + \frac{1}{2} \sum_{i=1}^{n} h(x_i, a_i) \\
&= \sum_{i=1}^{n} (\eta L(\sum_{s=1}^{t} \lambda_s) + 1/2) h(x_i, a_i)
\end{aligned}$$

The third inequality follows because $L(\lambda)$ is linear in λ and $h(x_i, a_i) \in \{0, 1\}$. Finally, we show even though the number of λ-variables is exponential in n, the algorithm can be efficiently implemented. In fact, the best response of the λ-player always returns a solution where all the variables are zero or exactly one is set to B. Therefore, instead of recording the entire λ vector the learning algorithm can just record the non-zero variables and there will be at most T of them.

ALGORITHM 4: Inner Optimization

Input: $\eta > 0$, weight $w^0 \in \mathbb{R}^n_+$, number of rounds T
Set $h_1 = 0$
for $t \in [T]$ **do**
  $\lambda_t = \text{Best}_\lambda(h_t)$
  Set $\widetilde{\lambda}_t = \sum_{t'=1}^{t} \lambda_{t'}$
  $h_{t+1} = \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} (\eta L_i(\widetilde{\lambda}_t) + 1/2) h(x_i, a_i)$
end
return *Uniform distribution over* $\{h_1, \ldots, h_T\}$.

Theorem 4. *Suppose* $|\ell(y, \widehat{y})| \leq M$ *for all* $y, \widehat{y}$. *Then algorithm 4 computes a* $(2M + B)\sqrt{n/T} + B(4\gamma_1 + \gamma_2)$*-approximate minmax equilibrium of the game* $U(h, \lambda)$ *for* $h \in \mathcal{H}$ *and* $\lambda \in \mathbb{R}_+^{|N(\gamma_1, \mathcal{W})| \times |\mathcal{A}|^2}$, $\|\lambda\|_1 \leq B$.

Proof. At round t, the cost is linear in h_t i.e. $f_t(h_t) = \sum_{i=1}^{n} L(\lambda_t)_i h_t(x_i, a_i)$. Let us write $\bar{\lambda} = \frac{1}{T}\lambda_t$ and D to be the uniform distribution over $h_1, \ldots, h_T$. Since we chose $R(x) = 1/2\|x\|_2^2$ as the regularization function and the actions are $0 - 1$ vectors in n-dimensional space, the diameter D_R is bounded by $\sqrt{n}$. On the other hand, $\|\nabla f_t(h_t)\|_\infty = \max_i |L(\lambda_t)_i|$. We now bound $|L(\lambda_t)_i|$ for an arbitrary i. Suppose $y_i = 1$. The proof when $y = 0$ is identical.

$$\begin{aligned}
|L(\lambda_t)_i| = \left|c_i^1 - c_i^0\right| &= \left|w_i^0\right| |\ell(0,1) - \ell(1,1)| + |\Delta_i| \\
&\leq 2M + B
\end{aligned}$$

The last line follows as $w_i^0 \leq 1$ and since λ_t is an approximate best reponse computed by algorithm 2, exactly one λ variable is set to B. Therefore, by theorem 3, for any hypothesis $h \in \mathcal{H}$,

$$\begin{aligned}
&\sum_{t=1}^{T} \sum_{i=1}^{n} L(\lambda_t)_i h_t(x_i, a_i) - \sum_{i=1}^{n} L(\lambda_t)_i h(x_i, a_i) \leq (2M + B)\sqrt{nT} \\
\Leftrightarrow &\sum_{t=1}^{T} U(h_t, \lambda_t) - U(h, \lambda_t) \leq (2M + B)\sqrt{nT} \\
\Leftrightarrow &\frac{1}{T} \sum_{t=1}^{T} U(h_t, \lambda_t) \leq U(h, \bar{\lambda}) + \frac{(2M + B)\sqrt{n}}{\sqrt{T}} \qquad (9)
\end{aligned}$$

On the other hand, λ_t is an approximate $B(4\gamma_1+\gamma_2)$-approximate best response to h_t for each round t. Therefore, for any λ we have,

$$\sum_{t=1}^{T} U(h_t,\lambda_t) \geq \sum_{t=1}^{T} U(h_t,\lambda) - BT(4\gamma_1+\gamma_2)$$
$$\Leftrightarrow \frac{1}{T}\sum_{t=1}^{T} U(h_t,\lambda_t) \geq \mathbb{E}_{h\sim D} U(h,\lambda) - B(4\gamma_1+\gamma_2) \tag{10}$$

Equations 9 and 10 immediately imply that the distribution D and $\bar{\lambda}$ is a $(2M+B)\sqrt{n/T}+B(4\gamma_1+\gamma_2)$-approximate equilibrium of the game $U(h,\lambda)$ ([7]). □

The next theorem establishes the guarantees of the approximate minmax solution. The proof is similar to the proof of theorem 4.5 from [10].

Theorem 5. *Let $(\widehat{h},\widehat{\lambda})$ be a ν-approximate minmax equilibrium of the game $U(h,\lambda)$. Then,*

$$\sum_{i=1}^{n} w_i^0 \ell(\widehat{h}(x_i,a_i),y_i) \leq \min_{h\in\mathcal{H}} \sum_{i=1}^{n} w_i^0 \ell(h(x_i,a_i),y_i) + 2\nu$$

and

$$\forall w\in\mathcal{W} \quad \delta_{DP}^{w}(\widehat{h}) \leq \varepsilon + \frac{M+2\nu}{B}$$

Proof. Let $(\widehat{h},\widehat{\lambda})$ be a ν-approximate minmax equilibrium of the game $U(h,\lambda)$ i.e.

$$\forall h \quad U(\widehat{h},\widehat{\lambda}) \leq U(h,\widehat{\lambda}) + \nu \quad \text{and} \quad \forall\lambda \quad U(\widehat{h},\widehat{\lambda}) \geq U(\widehat{h},\lambda) - \nu$$

Let h^* be the optimal feasible hypothesis. First suppose that $\widehat{h}$ is feasible i.e. $T(w,a,\widehat{h}) - T(w,a',\widehat{h}) \leq \varepsilon - 4\gamma_1$ for all $w\in N(\gamma_1,\mathcal{W})$ and $a,a'\in\mathcal{A}$. In that case, the optimal λ is the zero vector and $\max_\lambda U(\widehat{h},\lambda) = \sum_{i=1}^{n} w_i^0\ell(h(x_i,a_i),y_i)$. Therefore,

$$\sum_{i=1}^{n} w_i^0 \ell(\widehat{h}(x_i,a_i),y_i) = \max_{\lambda} U(\widehat{h},\lambda) \leq U(\widehat{h},\widehat{\lambda}) + \nu \leq U(h^*,\widehat{\lambda}) + 2\nu \leq \sum_{i=1}^{n} w_i^0 \ell(h^*(x_i,a_i),y_i) + 2\nu$$

The last inequality follows because h^* is feasible and λ is non-negative. Now consider the case when $\widehat{h}$ is not feasible i.e. there exists w,a,a' such that $T(w,a,\widehat{h}) - T(w,a',\widehat{h}) > \varepsilon - 4\gamma_1$. In that case, let $(\widehat{w},\widehat{a},\widehat{a}')$ be the tuple with maximum violation and the optimal λ, say λ^*, sets this coordinate to B and everything else to zero. Then

$$\sum_{i=1}^{n} w_i^0 \ell(\widehat{h}(x_i,a_i),y_i) = U(\widehat{h},\lambda^*) - B(T(\widehat{w},\widehat{a},\widehat{h}) - T(\widehat{w},\widehat{a}',\widehat{h}) - \varepsilon + 4\gamma_1)$$
$$\leq U(\widehat{h},\lambda^*) \leq U(\widehat{h},\widehat{\lambda}) + \nu \leq U(h^*,\widehat{\lambda}) + 2\nu \leq \sum_{i=1}^{n} w_i^0 \ell(h^*(x_i,a_i),y_i) + 2\nu.$$

The previous chain of inequalities also give

$$B\left(\max_{(w,a,a')} T(w,a,\widehat{h}) - T(w,a',\widehat{h}) - \varepsilon + 4\gamma_1\right) \leq \sum_{i=1}^{n} w_i^0 \ell(h^*(x_i,a_i),y_i) + 2\nu \leq M + 2\nu.$$

This implies that for all weights $w\in N(\gamma_1,\mathcal{W})$ the maximum violation of the fairness constraint is $(M+2\nu)/B$, which in turn implies a bound of at most $(M+2\nu)/B+\varepsilon$ on the fairness constraint with respect to any weight $w\in\mathcal{W}$. □

5 Faster Approximate Fair Classifier

6 Experiment

7 Conclusion

References

[1] Compas dataset. https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis. Accessed: 2019-10-26.

[2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. *arXiv preprint arXiv:1803.02453*, 2018.

[3] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.

[4] Joy Buolamwini and Timnit Gebru. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Conference on fairness, accountability and transparency*, pages 77–91, 2018.

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

[6] Robert S Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust Optimization for Non-Convex Objectives. In *Advances in Neural Information Processing Systems*, pages 4705–4714, 2017.

[7] Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *COLT*, volume 96, pages 325–332. Citeseer, 1996.

[8] Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of Opportunity in Supervised Learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

[9] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[10] 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.

[11] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning Fair Representations. In *International Conference on Machine Learning*, pages 325–333, 2013.