Fairness Checking

January 19, 2020

1 Introduction

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. These applications use learning algorithms trained on past data. However, past data is almost always biased in some way, and such bias is often reflected in the eventual decision. For example, Bolukbasi et al. [Bol+16] show that popular word embeddings implicitly encode societal biases, such as gender norms. Similarly, Buolamwini and Gebru [BG18] 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 [Zem+13; HPS+16; Aga+18]. 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 Model

We first check whether an algorithm is fair against a family of possible distributions. In particular, we consider distributions which are weighted empirical distributions and weights are chosen so that the new weighted distribution is close to the original training distribution.

2.1 Setup

Suppose, we have access to a "protected" attribute $A \in \{0, 1\}$, and a qualification attribute $Y \in \{0, 1\}$. In practice, a "protected" attribute might be race, gender, or some other attribute that might yield biased decision-making. Let us assume that $X \in \mathcal{X}$ denotes the set of remaining attributes which are used as input to a classifier f. For a distribution P over the space of attributes $\mathcal{X}$, we consider the following two fairness criteria:

- Demographic Parity (**DP**):

$$\left|\mathbb{E}_P[f(X, a)|A = a] - \mathbb{E}_P[f(X, a')|A = a']\right| \leq \epsilon$$

  for all a and a'.

- Equalized Odds (**EO**):

$$\left|\mathbb{E}_P[f(X, a)|Y = y, A = a] - \mathbb{E}_P[f(X, a')|Y = y, A = a']\right| \leq \epsilon$$

  for all y, a, and a'.

We assume that we have data (X_i, Y_i, A_i) for $i = 1, \ldots, n$ and P can be represented as a weighted empirical distribution i.e. for any $(x, y, a) \in \mathcal{X} \times \{0, 1\} \times \{0, 1\}$ we have:

$$P(x, y, a) = \sum_{i=1}^{n} w_i \mathbf{1}_{(X_i, Y_i, A_i)=(x,y,a)}.$$

where the weights are specified with a weight vector $w = (w_1, ..., w_n)$ such that $w_i \geq 0$ for all i and $\sum_i w_i = 1$.

The two fairness definitions above are fairly standard and well-known in the fair ML literature. Both are examples of "group fairness." Demographic Parity, also known as Statistical Parity or Independence, [Dwo+11] means that the difference in positive rates for the two groups ("protected" and "unprotected") differ by some small ϵ. Equalized Odds, also called Positive Rate Parity or Separation, requires the two groups' true positive and false positive rates differ by some small ϵ.

2.2 Checking Demographic Parity (DP)

As a start, we assume that the weighted empirical distributions are such that the marginal distributions over the protected attributes are preserved. In particular, we consider weights such that $\sum_{i=1}^{n} w_i \mathbf{1}_{A_i=a} = \pi_a$ for all a. Here the π_a are the proportions for the protected attributes, which we assume are known. We use the following linear programs to check if the classifier f fails the fairness criterion on some allowable weighted empirical distribution. For each a, a':

$$\begin{aligned}
\max_{w} \quad & \frac{1}{\pi_a} \sum_{i=1}^{n} w_i f(X_i) \mathbf{1}_{A_i=a} - \frac{1}{\pi_{a'}} \sum_{i=1}^{n} w_i f(X_i) \mathbf{1}_{A_i=a'} \\
\text{s.t.} \quad & \sum_{i=1}^{n} w_i \mathbf{1}_{A_i=a} = \pi_a \\
& \sum_{i=1}^{n} w_i \mathbf{1}_{A_i=a'} = \pi_{a'} \\
& w_i \geq 0 \quad \forall i \in [n] \\
& \sum_{i=1}^{n} w_i = 1
\end{aligned} \tag{1}$$

If there exists a pair of protected attributes a, a' such that the optimal value of the linear program is more than ϵ, then we have found a violation of DP.

2.3 Checking Equalized Odds (EO)

As in the previous section, we assume that we only care about weighted empirical distributions such that $\sum_{i=1}^{n} w_i \mathbf{1}_{A_i=a, Y_i=y} = \pi_{a,y}$ for all a and y where $\pi_{a,y}$ are known proportions for the protected and qualification attributes. We again use the following linear programs to check if f fails the fairness criterion on some

weighted empirical distribution.

$$\begin{aligned}
\max_{w} \quad & \frac{1}{\pi_{a,y}} \sum_{i=1}^{n} w_i f(X_i) \mathbf{1}_{A_i=a, Y_i=y} - \frac{1}{\pi_{a',y}} \sum_{i=1}^{n} w_i f(X_i) \mathbf{1}_{A_i=a', Y_i=y} \\
\text{s.t.} \quad & \sum_{i=1}^{n} w_i \mathbf{1}_{A_i=a, Y_i=y} = \pi_{a,y} \\
& \sum_{i=1}^{n} w_i \mathbf{1}_{A_i=a', Y_i=y} = \pi_{a',y} \\
& w_i \geq 0 \quad \forall i \in [n] \\
& \sum_{i=1}^{n} w_i = 1
\end{aligned} \tag{2}$$

Like in DP, if there exists a pair of protected attributes a, a' such that the optimal value of the linear program is more than ϵ, then we have found a violation of DP.

3 Evaluation

Using the above framework, we evaluated the two group fairness properties (DP and EO) on two real world datasets used frequently in the fairness literature: Adult [Adu] and COMPAS [Com]. We evaluated the robustness of classifiers satisfying DP and/or EO to perturbations in the original training distributions, given by some weighted empirical distribution. First, we trained fair classifiers on each of the datasets using well-known preprocessing techniques to achieve Demographic Parity and Equalized Odds within an acceptable ϵ. Then, we allowed the weights of the empirical distribution to vary within a small margin of ϵ. Taking the first empirical weighting such that $\epsilon \geq 0.10$, we analyzed the new marginal distributions of the data and compared it with the original marginal distribution of the data. We found that very small changes in the marginal distributions of the data led to the classifier violating ϵ, suggesting the existing fair classifiers are not robust.

3.1 Datasets

In our experiments, we use two real-world datasets: Adult and COMPAS. The Adult dataset consists of 14 attributes (e.g. age, education level, etc.) and 48,842 instances, used for predicting whether income exceeds \$50K/year based on U.S. Census data. The binary label (Y) in this dataset is positive if the subject's income exceeds \$50K/year USD and is negative if the subject's income is less than \$50K/year USD. For this dataset, we consider sex as the binary protected attribute (A), which is either Male ($A = 1$) or Female ($A = 0$). The COMPAS dataset consists of 53 attributes (e.g. race, age, prior offenses) and 7,214 instances, used for predicting whether a criminal defendant will recidivate. The binary label in this dataset is positive if the subject recidivated after two years and negative if they did not recidivate. For COMPAS, we consider race as the binary protected attribute, which is either Caucasian or not Caucasian.

For the rest of this paper, we refer to the sex in the Adult dataset and race in the COMPAS dataset as the protected attribute, denoted by A. For the Adult dataset, we refer to Male as privileged class and Female as the non-privileged class, taking on values $A = 1$ and $A = 0$, respectively. For the COMPAS dataset, we refer to Caucasian as the privileged class and not Caucasian (e.g. African-American, Hispanic, etc.) as the non-privileged class, taking on values $A = 1$ and $A = 0$, respectively.

3.2 Experimental Setup

First, after doing standard preprocessing on the data (removing missing rows, feature selection, etc.) down to 45,222 instances for Adult and 6,172 instances for COMPAS, we trained logistic regression (LR) classifiers

on each by using the Optimized Pre-processing algorithm proposed by Calmon et al. [Cal+17]. Optimized Pre-processing uses a probabilistic framework that determines an optimal random mapping of the training dataset into a transformed dataset used to train the model. This method is model agnostic because it is a preprocessing technique and is shown by Calmon et. al. to perform competitively well compared to other fair preprocessing algorithms in the literature with respect to group fairness. In our implementation, this preprocessing algorithm achieves Demographic Parity and Equalized Odds as desired on both datasets, while maintaining a reasonable classification accuracy. We use δ_{DP} and δ_{EO} to denote the unfairness gap for DP and EO respectively.

$$\left|\mathbb{E}_P[f(X,a)|A=a] - \mathbb{E}_P[f(X,a')|A=a']\right| := \delta_{DP} \tag{3}$$

$$\left|\mathbb{E}_P[f(X,a)|Y=1,A=a] - \mathbb{E}_P[f(X,a')|Y=1,A=a']\right| := \delta_{EOY1} \tag{4}$$

$$\left|\mathbb{E}_P[f(X,a)|Y=0,A=a] - \mathbb{E}_P[f(X,a')|Y=0,A=a']\right| := \delta_{EOY0} \tag{5}$$

For COMPAS, standard logistic regression classifier achieved $\delta_{DP} = 0.17, \delta_{EOY0} = 0.12$, and $\delta_{EOY1} = 0.12$ with an accuracy of 0.66; whereas the optimized pre-processing algorithm proposed by [Cal+17] achieved $\delta_{DP} = 0.02, \delta_{EOY0} = 0.09$, and $\delta_{EOY1} = 0.05$ with an accuracy of 0.64. For Adult, standard logistic regression classifier achieved $\delta_{DP} = 0.21, \delta_{EOY0} = 0.11$, and $\delta_{EOY1} = 0.46$ with an accuracy of 0.81; and the Optimized Pre-processing achieved $\delta_{DP} = 0.06, \delta_{EOY0} = 0.01$, and $\delta_{EOY1} = 0.03$ with an accuracy of 0.79. Because, the Optimized Pre-processing achieved an "unfairness gap" of under 0.1 with a minimal reduction in accuracy, we use this classifier as the base fair classifier for our experiments.

After running the Optimized Pre-processing algorithm to train a fair logistic regression classifier on Adult and COMPAS with respect to Demographic Parity and Equalized Odds, we make use of our linear programs in Section 2.2 and Section 2.3. For both Demographic Parity and Equalized Odds, we follow the constraints in (1) and (2), but we add one more constraint:

$$\frac{1-\gamma}{n} \leq w_i \leq \frac{1+\gamma}{n} \tag{6}$$

where $\gamma \in (0,1)$ is a parameter we use to set how much or how little w_i can vary in our linear program. Note that, at $\gamma = 0$, we simply have $w_i = \frac{1}{n}$, the original empirical distribution on the data. The parameter γ allows us to control the distance between the weighted empirical distribution and the original distribution. Note that if the constraint eq. (6) is satisified by all the training instances, the L1 norm between the weighted empirical distribution and the original distribution is at most γ.

Demographic Parity: We aim to find the a value of γ where our weighted empirical distribution first violates Demographic Parity with $\epsilon > 0.1$. That is, by loosening the γ in the bound in (6) on w_i, we aim to find the first γ such that our objective function violates:

$$\max_w \quad \frac{1}{\pi_a}\sum_{i=1}^{n} w_i f(X_i)\mathbf{1}_{A_i=a} - \frac{1}{\pi_{a'}}\sum_{i=1}^{n} w_i f(X_i)\mathbf{1}_{A_i=a'} > 0.1 = \epsilon \tag{7}$$

subject to all the constraints before. Particularly, we test $\gamma \in \{0, 0.01, 0.02, ..., 0.98, 0.99, 1.0\}$. Then, on the distribution of w_i where $\epsilon > 0.1$ for our objective function, we analyze the marginal distributions on the attributes of our new, "unfair" empirical distribution. We measure the differences between the marginal distributions of the original distribution and the weighted unfair distributions to get an idea how different the distributions are, and how robust the fair classifiers are to perturbations in the training distribution. These results are detailed in Section 3.3.

Equalized Odds: We also aim to find a value of γ where our weighted empirical distribution first violates Equalized Odds with $\epsilon > 0.1$. In the context of Equalized Odds, this means the following:

$$\max_w \quad \frac{1}{\pi_{a,0}}\sum_{i=1}^{n} w_i f(X_i)\mathbf{1}_{A_i=a,Y_i=0} - \frac{1}{\pi_{a',0}}\sum_{i=1}^{n} w_i f(X_i)\mathbf{1}_{A_i=a',Y_i=0} > 0.1 = \epsilon \tag{8}$$

$$\max_w \quad \frac{1}{\pi_{a,1}}\sum_{i=1}^{n} w_i f(X_i)\mathbf{1}_{A_i=a,Y_i=1} - \frac{1}{\pi_{a',1}}\sum_{i=1}^{n} w_i f(X_i)\mathbf{1}_{A_i=a',Y_i=1} > 0.1 = \epsilon \tag{9}$$

where (8) represents the difference in false positive rates ($Y = 0, f(X) = 1$) and (9) represents the difference in true positive rates ($Y = 1, f(X) = 1$), and the constraints for Equalized Odds are as before. Like in Demographic Parity, we test γ in the range $\{0, 0.01, 0.02, ..., 0.98, 0.99, 1.0\}$ and find a distribution of w_i where $\epsilon > 0.1$. We compare this new, "unfair" empirical distribution through its marginal distributions to the original, "fair" empirical distribution. These results are detailed in Section 3.3.

3.3 Results

We measure the $L1$-distance between the marginal distributions over the attributes for the original distribution and the weighted "unfair" weighted distribution. Because each feature was binary, the L1 distance for attribute x was simply given by:

$$D_x = \left|\Pr_w[x = 1] - \Pr_o[x = 1]\right| + \left|\Pr_w[x = 0] - \Pr_o[x = 0]\right| \tag{10}$$

where $\Pr_w[x = n]$ is the marginal distribution from the reweighted, violating distribution and $\Pr_o[x = n]$ is the marginal distribution from the original, fair distribution. In Table 1 and Table 2 below, we summarize the results for a selected subset of attributes for each experiment.

COMPAS	γ	sex	age_25	age_25_45	age_45	priors_0	priors_1_3	priors_3	Avg. Dist.
DP	0.14	0.0014	0.0018	0.0042	0.0060	0.0010	0.015	0.025	0.0090
EO, Y = 0	0.32	0.0040	0.016	0.0089	0.025	0.0065	0.020	0.014	0.015
EO, Y = 1	0.19	0.0062	0.0053	0.0014	0.0038	0.013	0.0067	0.020	0.0071

Table 1: L1 distance from fair to "unfair" marginal distributions for COMPAS. γ is the smallest value of γ in constraint (6) to make LP difference (unfairness) ≥ 0.1. Selected attributes on display: `sex` is the sex of the individual, `age_n` attributes are age $<$ 25, 25 $<$ age $<$ 45, and age $>$ 45, `priors_n` means prior crimes = 0, 1 $<$ prior crimes $<$ 3, and 3 $<$ prior crimes. Avg. Dist. is the average L1 distance through *all* the features (some not shown here).

Adult	γ	race	age_20	age_40	age_60	edu6	edu8	edu10	edu12	Avg. Dist.
SP	0.49	0.0049	0.021	0.017	0.0012	0.0055	0.0015	0.031	0.0052	0.014
EO (Y = 0)	0.47	0.0019	0.0060	0.00051	0.00064	0.0027	0.00064	0.011	0.00039	0.0033
EO (Y = 1)	0.17	0.0021	0.0021	0.0026	0.00042	0.00079	0.00051	0.012	0.00135	0.0036

Table 2: L1 distance from fair to "unfair" marginal distributions for Adult. γ is the smallest value of γ in constraint (6) to make LP difference (unfairness) ≥ 0.1. Selected attributes on display: `race` is the race of the individual (binary, White or not White), `age_n` attributes are 20 $<$ age $<$ 30, 40 $<$ age $<$ 50, and 60 $<$ age $<$ 70, `edu_n` means years of education total. Avg. Dist. is the average L1 distance through *all* the features (some not shown here).

We observe that, by modifying the original "fair" distribution with uniform weights to a weighted empirical distribution that violates fairness for both properties (with gap ≥ 0.1), the distributions are, in fact, extremely similar. The average L1 distance in marginal distributions between features in the original distribution and the violating distribution are all within the range of 0 and 1.5%. For COMPAS DP, the maximum L1 difference in marginal distributions was 2.5% for `priors_3` and for COMPAS EO, the maximum L1 differences were 2.5% (`age_45`, $Y = 0$) and 2.0% (`priors_3`, $Y = 1$). For Adult DP, the maximum L1 difference in marginal distributions was 8.8% (`edu_>12`, not shown above) and for COMPAS EO, the max L1 differences were 1.7% (`edu_>12`, $Y = 0$) and 2.6 % (`edu_>12`, $Y = 1$). In Appendix A, we include histograms for the marginal distributions of each of the attributes, including those not shown in the tables above. Therefore, we observe that small changes in the weighted empirical distribution of the data result in a violation of unfairness, bringing into question the robustness of fair classifiers.

4 Robust and Fair Classification

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.

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 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:

$$\min_{h \in \mathcal{H}_\mathcal{W}} \max_{w \in \mathcal{W}} \ell(h, w) \tag{11}$$

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 $\ell(Q, w) = \sum_h Q(h)\ell(h, w)$.

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 11. The algorithm is based on ideas presented in [Che+17], 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 Chen et al. [Che+17]. □

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

$$\frac{\sum_{i:a_i=a} w_i f(x_i, a)}{\sum_{i:a_i=a} w_i} \leq (1+\gamma_1) \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i - \gamma_1/2} \leq \frac{1+\gamma_1}{1-\gamma_1} \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i} \leq (1+3\gamma_1) \frac{\sum_{i:a_i=a} w'_i f(x_i, a)}{\sum_{i:a_i=a} w'_i} \tag{12}$$

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 $\delta^w_{DP}(f) = \sup_{a,a'} |T(w, a, f) - T(w, a', f)|$. Now our aim is to solve the following problem.

$$\begin{aligned}&\min_{h\in\mathcal{H}} \sum_{i=1}^n w^0_i \ell(h(x_i, a_i), y_i) \\ &\text{s.t.} T(w, a, h) - T(w, a', h) \leq \varepsilon - 4\gamma_1 \ \forall w \in N(\gamma_1, \mathcal{W}) \ \forall a, a' \in \mathcal{A}\end{aligned} \tag{13}$$

We form the following Lagrangian.

$$\min_{h\in\mathcal{H}} \max_{\substack{\lambda\in\mathbb{R}_+^{|N(\gamma_1,\mathcal{W})|\times|\mathcal{A}|^2} \\ \|\lambda\|_1\le B}} \sum_{i=1}^{n} w_i^0 \ell(h(x_i,a_i),y_i) + \sum_{w\in N(\gamma_1,\mathcal{W})} \sum_{a,a'\in\mathcal{A}} \lambda_w^{a,a'} (T(w,a,h) - T(w,a',h) - \varepsilon + 4\gamma_1) \quad (14)$$

We now focus on solving the problem defined in equation 14. In order to do so, we first convert equation 14 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.

$$c_i^0 = \begin{cases} \ell(0,1)w_i^0 & \text{if } y_i = 1 \\ \ell(0,0)w_i^0 & \text{if } y_i = 0 \end{cases} \qquad c_i^1 = \begin{cases} \ell(1,1)w_i^0 + \Delta_i & \text{if } y_i = 1 \\ \ell(1,0)w_i^0 + \Delta_i & \text{if } y_i = 0 \end{cases} \quad (15)$$

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

$$\widehat{h} \in \arg\min_{h\in\mathcal{H}} \sum_{i=1}^{n} \left\{c_i^1 h(x_i,a_i) + c_i^0 (1 - h(x_i,a_i))\right\} \quad (16)$$

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 $\mathrm{val}(a, a', \pi) = \frac{1}{\pi_a} \sum_{i:a_i=a} w(a, a', \pi)_i h(x_i, a) - \frac{1}{\pi_{a'}} \sum_{i:a_i=a'} w(a, a', \pi)_i h(x_i, a')$
  end
end
Set $(a^*, a'^*, \pi^*) = \arg\max_{a,a',\pi} \mathrm{val}(a, a', \pi)$
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 $\lambda_w^{a,a'} = \begin{cases} B & \text{if } (a, a', w) = (a^*, a'^*, w') \\ 0 & o.w. \end{cases}$
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 14. 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. [Haz+16]).

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.

$$\widehat{h} \in \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} c_i^1(\lambda) h(x_i, a_i) + c_i^0(\lambda)(1 - h(x_i, a_i)) \tag{17}$$

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

$$\widehat{h} \in \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} L_i(\lambda) h(x_i, a_i) \tag{18}$$

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 13. 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* $|\ell(y, \widehat{y})| \le M$ *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* $\lambda \in \mathbb{R}_+^{|N(\gamma_1,\mathcal{W})|\times|\mathcal{A}|^2}$, $\|\lambda\|_1 \le B$.

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.

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

The last line follows as $w_i^0 \le 1$ 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 $h \in \mathcal{H}$,

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{n} L(\lambda_t)_i h_t(x_i, a_i) - \sum_{i=1}^{n} L(\lambda_t)_i h(x_i, a_i) \leq (2M + B)\sqrt{nT} \\
\Leftrightarrow &\sum_{t=1}^{T} U(h_t, \lambda_t) - U(h, \lambda_t) \leq (2M + B)\sqrt{nT} \\
\Leftrightarrow &\frac{1}{T}\sum_{t=1}^{T} U(h_t, \lambda_t) \leq U(h, \bar{\lambda}) + \frac{(2M + B)\sqrt{n}}{\sqrt{T}} \qquad (19)
\end{aligned}$$

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,

$$\begin{aligned}
&\sum_{t=1}^{T} U(h_t, \lambda_t) \geq \sum_{t=1}^{T} U(h_t, \lambda) - BT(4\gamma_1 + \gamma_2) \\
\Leftrightarrow &\frac{1}{T}\sum_{t=1}^{T} U(h_t, \lambda_t) \geq \mathbb{E}_{h\sim D} U(h, \lambda) - B(4\gamma_1 + \gamma_2) \qquad (20)
\end{aligned}$$

Equations 19 and 20 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 $U(h, \lambda)$ ([FS96]). □

The next theorem establishes the guarantees of the approximate minmax solution. The proof is similar to the proof of theorem 4.5 from [Kea+17].

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

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

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 Conclusion

In this paper, we introduce tools to verify whether an algorithm satisfies fairness for the well-known group fairness properties of Demographic Parity and Equalized Odds. We run experiments with these tools on simple weighted empirical distributions of the data. By allowing the weights of the distribution to change within a small γ window, we found γ such that our new distribution violates fairness by $\epsilon = 0.1$ Upon comparing the marginal distributions of the "unfair," re-weighted distribution to the original "fair" distribution, we found small differences. That is, with small perturbations to the distribution of data, we can get our classifier to violate group fairness. There are several directions for future work.

1. An immediate next step is to run our experiment on other fair classifiers and test how robust they are. Since we already ran our experiment with a pre-processing classifier [Cal+17], it will be interesting to see how an in-processing classifier e.g. [Aga+18], or a post-processing classifier e.g. [HPS+16] performs.

2. Our experiments bring into question the design of fair classifiers that are robust to perturbations in the training set. Like Hardt, Price, and Srebro [HPS+16], it would be nice to find a post-processing step that makes the classifiers robust, as it would avoid re-training the whole classifier from scratch.

3. If the post-processing approach fails, we can use the framework developed by Agarwal et al. [Aga+18] who designs an in-processing classifier by converting the problem of maximizing accuracy subject to fairness constraints to a sequence of cost-sensitive classification problems. We can add the robustness constraints or some convex analogues of them in addition to the fairness constraints and check if the whole framework still works or not. Another interesting direction would be to leverage the rich literature on distributionally robust optimization (DRO) [ND16]. The DRO framework mainly works with unconstrained classifiers and it would be interesting to see if we can incorporate the fairness constraints in this framework.

4. It would also be interesting to extend our framework so that it works even when the sensitive attribute is not explicitly given. One example of this case is the problem of facial recognition. Buolamwini and Gebru [BG18] discovered biases in the existing facial recognition systems and highlighted the need for fairness check in such situation. Note that, one might argue that we can just train a classifier that predicts the sensitive attribute and then use our framework. However, the problem is that such classifier is trained on biased data and it will inevitably show different accuracies for different sensitive groups.

5. Finally, Kearns et al. [Kea+17] developed classifiers that guarantees fairness with respect to a large class of subgroups, not just a fixed class of pre-specified subgroups. An interesting direction of future work would be to develop tools that can detect small subgroups which are discriminated. Notice that, a naive extension of our LP based framework will not work, as the possible number of subgroups can be exponentially large in the number of features.

References

[Adu] *Adult Dataset*. https://archive.ics.uci.edu/ml/datasets/Adult. Accessed: 2019-10-26 (cit. on p. 3).

[Aga+18] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. "A reductions approach to fair classification". In: *arXiv preprint arXiv:1803.02453* (2018) (cit. on pp. 1, 13).

[BG18] Joy Buolamwini and Timnit Gebru. "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification". In: *Conference on fairness, accountability and transparency*. 2018, pp. 77–91 (cit. on pp. 1, 13).

[Bol+16] 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*. 2016, pp. 4349–4357 (cit. on p. 1).

[Cal+17] 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*. 2017, pp. 3992–4001 (cit. on pp. 4, 13).

[Che+17] Robert S Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. "Robust Optimization for Non-Convex Objectives". In: *Advances in Neural Information Processing Systems*. 2017, pp. 4705–4714 (cit. on pp. 6, 7).

[Com] *COMPAS Dataset*. https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis. Accessed: 2019-10-26 (cit. on p. 3).

[Dwo+11] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. "Fairness Through Awareness". In: *CoRR* abs/1104.3913 (2011). arXiv: 1104.3913. URL: http://arxiv.org/abs/1104.3913 (cit. on p. 2).

[FS96] Yoav Freund and Robert E Schapire. "Game theory, on-line prediction and boosting". In: *COLT*. Vol. 96. Citeseer. 1996, pp. 325–332 (cit. on p. 12).

[Haz+16] Elad Hazan et al. "Introduction to online convex optimization". In: *Foundations and Trends® in Optimization* 2.3-4 (2016), pp. 157–325 (cit. on p. 10).

[HPS+16] Moritz Hardt, Eric Price, Nati Srebro, et al. "Equality of Opportunity in Supervised Learning". In: *Advances in neural information processing systems*. 2016, pp. 3315–3323 (cit. on pp. 1, 13).

[Kea+17] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. "Preventing fairness gerrymandering: Auditing and learning for subgroup fairness". In: *arXiv preprint arXiv:1711.05144* (2017) (cit. on pp. 12, 13).

[ND16] Hongseok Namkoong and John C Duchi. "Stochastic gradient methods for distributionally robust optimization with f-divergences". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2208–2216 (cit. on p. 13).

[Zem+13] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. "Learning Fair Representations". In: *International Conference on Machine Learning*. 2013, pp. 325–333 (cit. on p. 1).

Appendix A

5.1 COMPAS DP Marginal Distributions

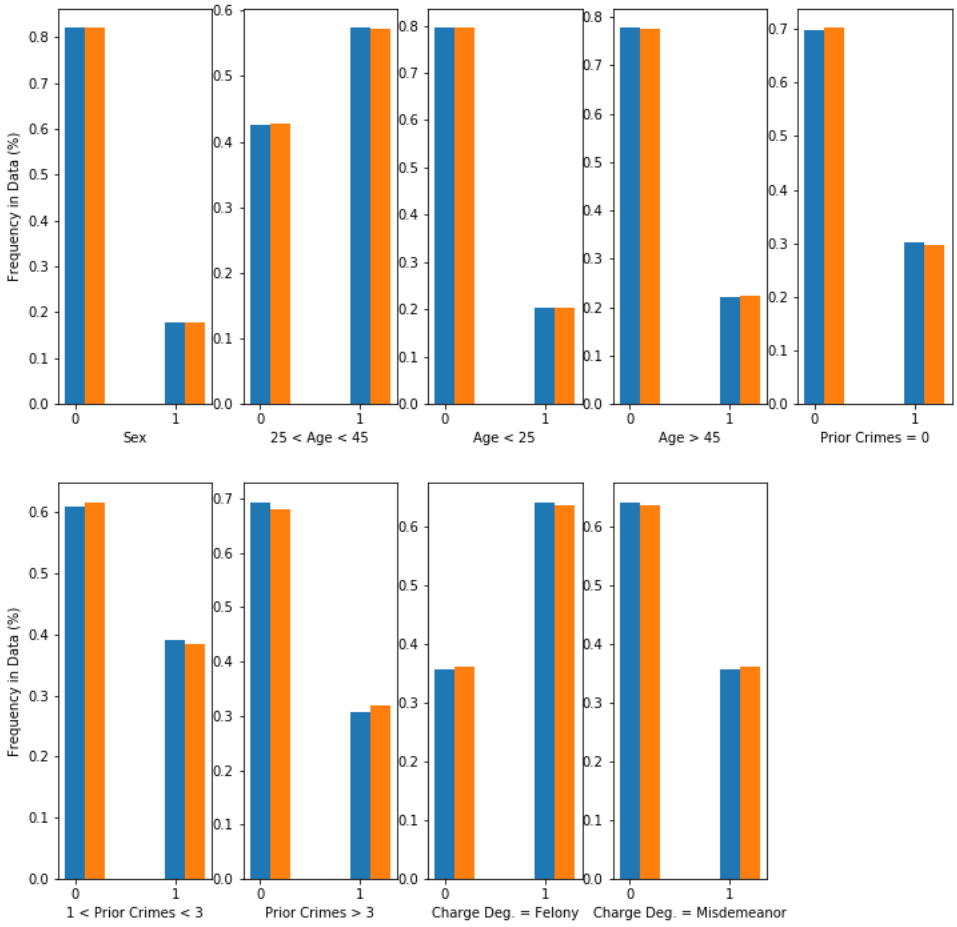

Figure 1: COMPAS DP marginal distributions. Blue is unweighted ("fair"), orange is reweighted ("unfair").

5.2 COMPAS EO Marginal Distributions

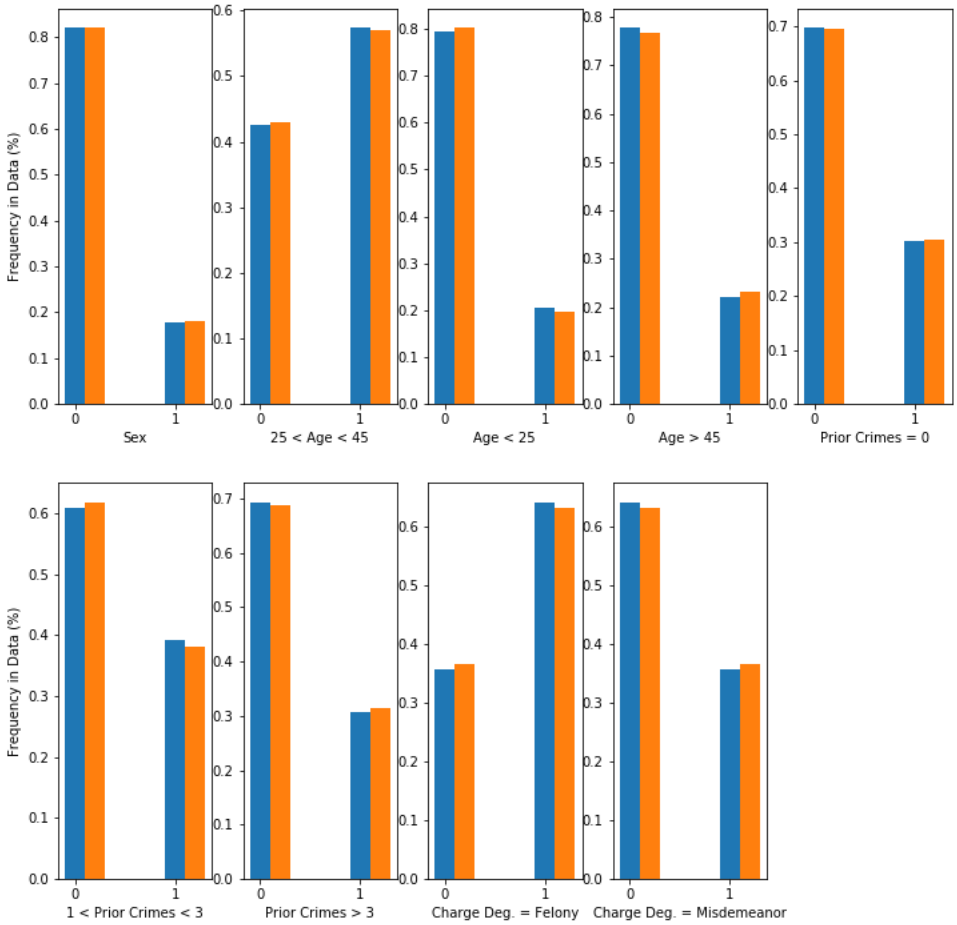

Figure 2: COMPAS EO marginal distributions for $Y = 0$. Blue is unweighted ("fair"), orange is reweighted ("unfair").

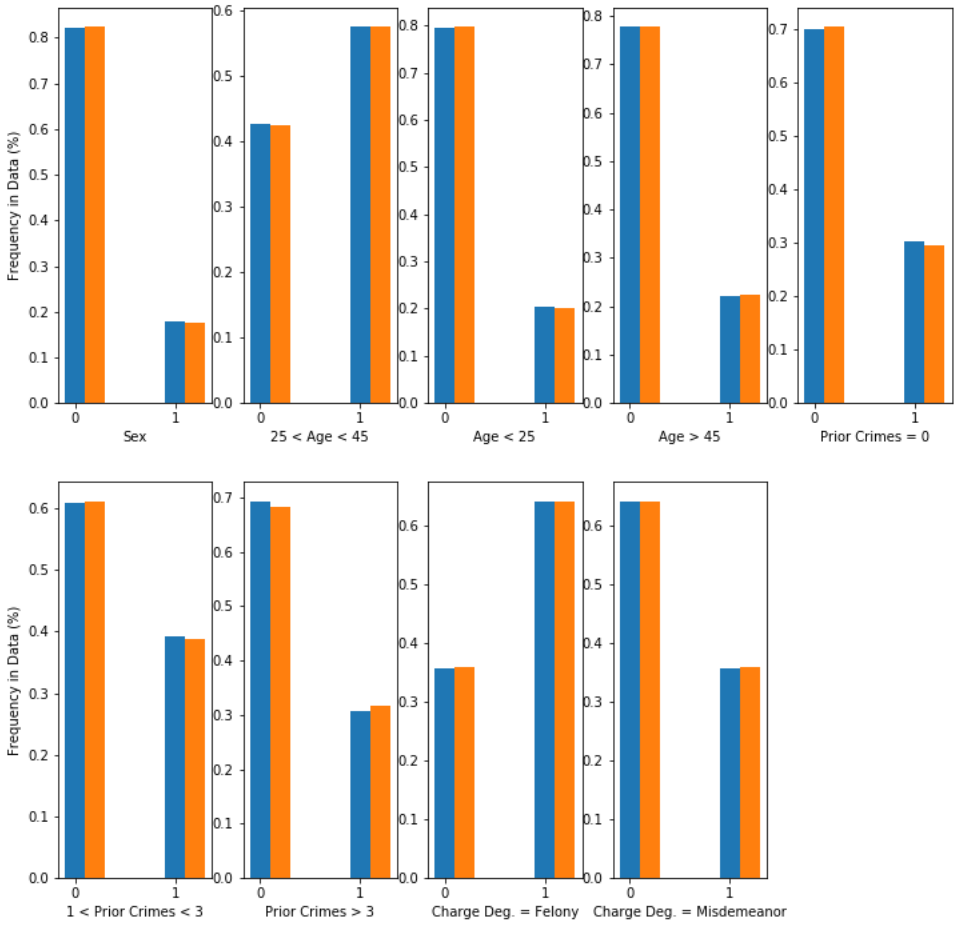

Figure 3: COMPAS EO marginal distributions for $Y = 1$. Blue is unweighted ("fair"), orange is reweighted ("unfair").

5.3 Adult DP Marginal Distributions

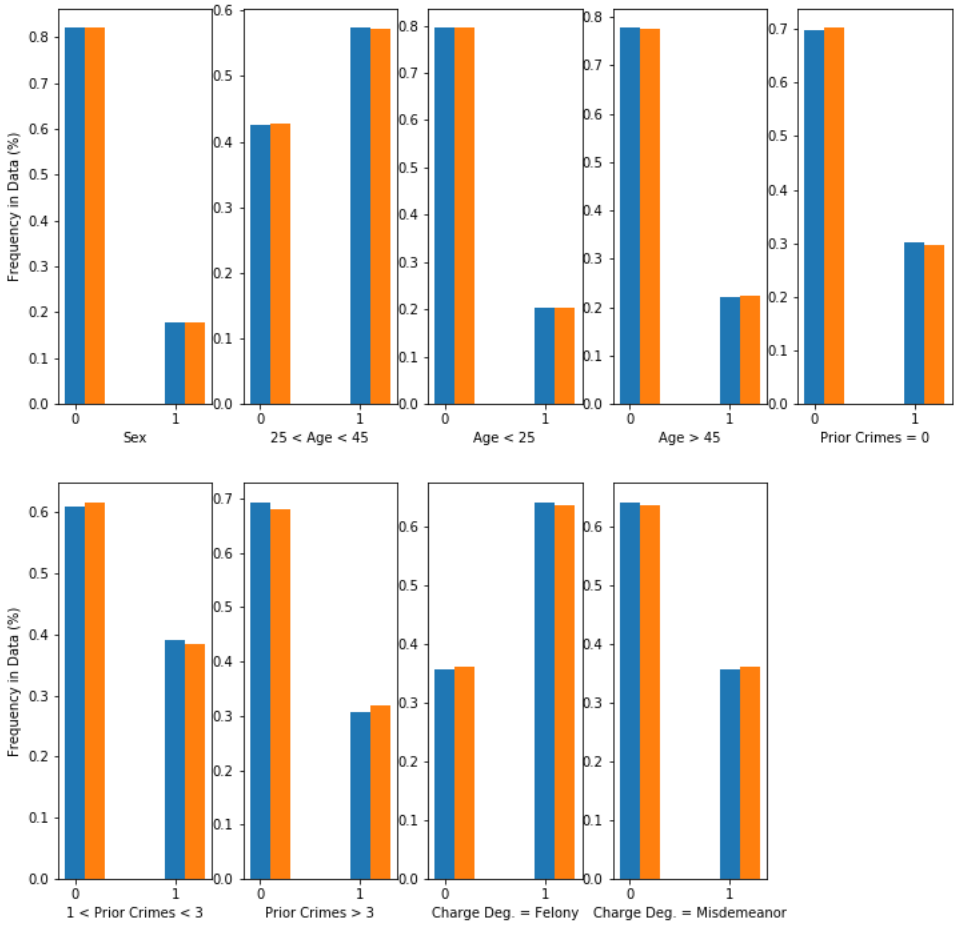

Figure 4: Adult DP marginal distributions. Blue is unweighted ("fair"), orange is reweighted ("unfair").

5.4 Adult EO Marginal Distributions

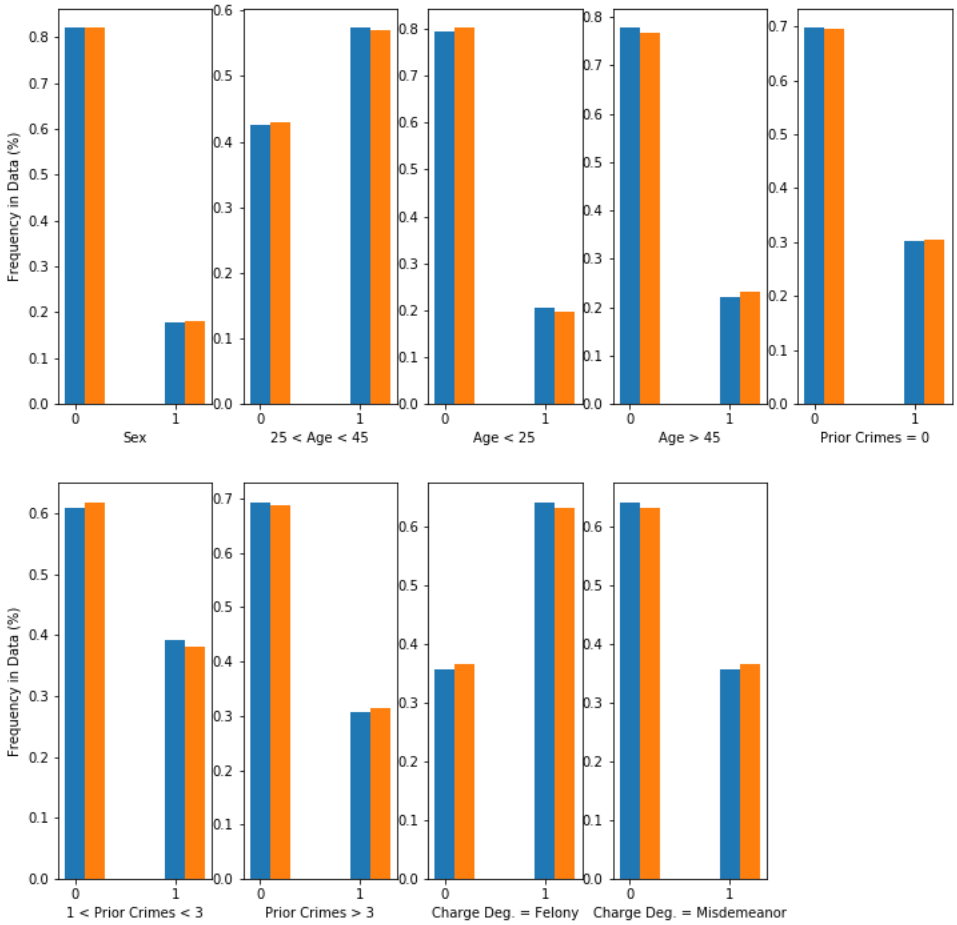

Figure 5: Adult EO marginal distributions for $Y = 0$. Blue is unweighted ("fair"), orange is reweighted ("unfair").

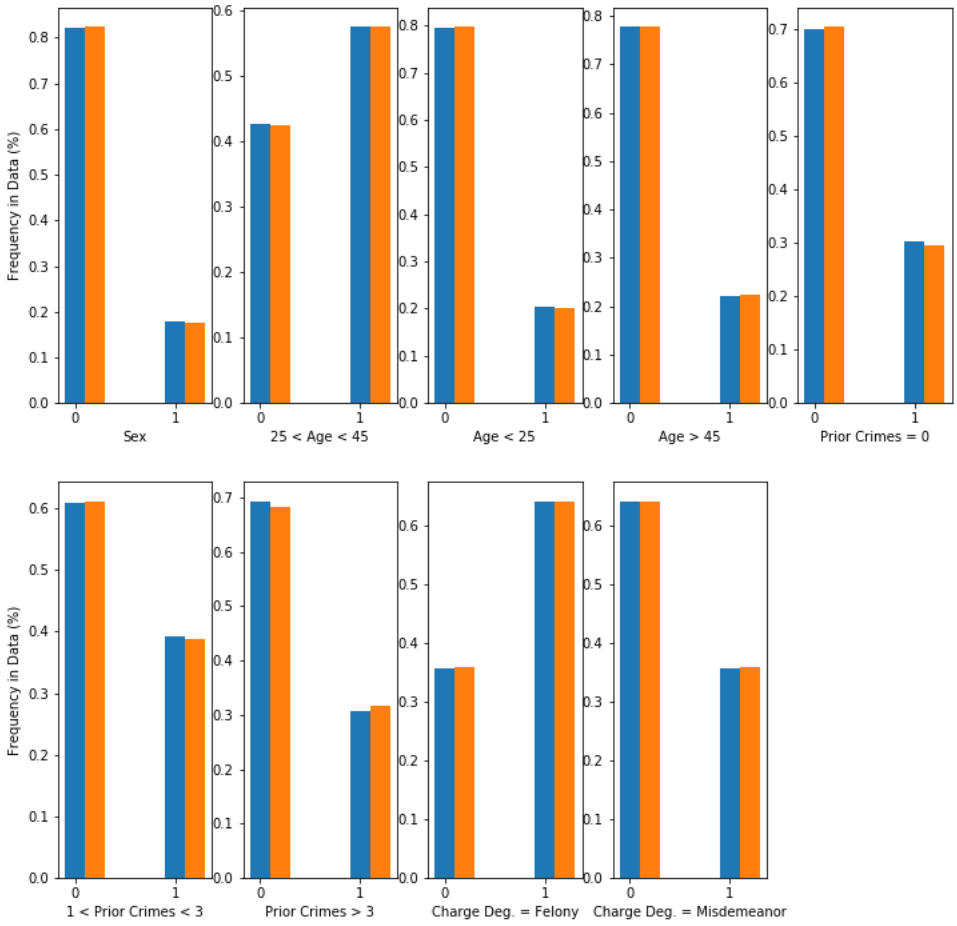

Figure 6: Adult EO marginal distributions for $Y = 1$. Blue is unweighted ("fair"), orange is reweighted ("unfair").