Fair Adversarial Gradient Tree Boosting

Vincent Grari
Sorbonne Université
LIP6/CNRS
Paris, France
vincent.grari@lip6.fr

Boris Ruf
AXA
REV Research
Paris, France
boris.ruf@axa.com

Sylvain Lamprier
Sorbonne Université
LIP6/CNRS
Paris, France
sylvain.lamprier@lip6.fr

Marcin Detyniecki
AXA
REV Research
Paris, France
marcin.detyniecki@axa.com

***Abstract*—Fair classification has become an important topic in machine learning research. While most bias mitigation strategies focus on neural networks, we noticed a lack of work on fair classifiers based on decision trees even though they have proven very efficient. In an up-to-date comparison of state-of-the-art classification algorithms in tabular data, tree boosting outperforms deep learning [1]. For this reason, we have developed a novel approach of adversarial gradient tree boosting. The objective of the algorithm is to predict the output Y with gradient tree boosting while minimizing the ability of an adversarial neural network to predict the sensitive attribute S. The approach incorporates at each iteration the gradient of the neural network directly in the gradient tree boosting. We empirically assess our approach on 4 popular data sets and compare against state-of-the-art algorithms. The results show that our algorithm achieves a higher accuracy while obtaining the same level of fairness, as measured using a set of different common fairness definitions.**

***Index Terms*—Fair machine learning, Adversarial learning, Gradient Tree Boosting**

I. Introduction

Since many applications of machine learning models have far-reaching consequences on people (credit approval, recidivism score etc.), there is growing concern about their potential to reproduce discrimination against a particular group of people based on sensitive characteristics such as gender, race, religion, or other. Many bias mitigation strategies for machine learning have been proposed in recent years, however, most of them focus on neural networks. Ensemble methods combining several decision tree classifiers have proven very efficient for various applications. Therefore, in practice for tabular data sets, actuaries and data scientists prefer the use of gradient tree boosting over neural networks due to its generally higher accuracy rates. Our field of interest is the development of fair classifiers based on decision trees. In this paper, we propose a novel approach to combine the strength of gradient tree boosting with an adversarial fairness constraint. The contributions of this paper are as follows:

- We apply adversarial learning for fair classification on decisions trees;
- We empirically compare our proposal and its variants with several state-of-the-art approaches, for two different fairness metrics. Experiments show the great performance of our approach.

II. Fair Machine Learning

A. Definitions of Fairness

Throughout this document, we consider a classical supervised classification problem training with n examples $(x_i, s_i, y_i)_{i=1}^n$, where $x_i \in \mathbf{R}^p$ is the feature vector with p predictors of the i-th example, s_i is its binary sensitive attribute and y_i its binary label.

In order to achieve fairness, it is essential to establish a clear understanding of its formal definition. In the following we outline the most popular definitions used in recent research. First, there is information sanitization which limits the data that is used for training the classifier. Then, there is individual fairness, which binds at the individual level and suggests that fairness means that similar individuals should be treated similarly. Finally, there is statistical or group fairness. This kind of fairness partitions the world into groups defined by one or several high level sensitive attributes. It requires that a specific relevant statistic about the classifier is equal across those groups. In the following, we focus on this family of fairness measures and explain the most popular definitions of this type used in recent research.

1) Demographic Parity: Based on this definition, a classifier is considered fair if the prediction $\widehat{Y}$ from features X is independent from the protected attribute S [2].

Definition 1. $P(\widehat{Y}=1|S=0) = P(\widehat{Y}=1|S=1)$

There are multiple ways to assess this objective. The p-rule assessment ensures the ratio of the positive rate for the unprivileged group is no less than a fixed threshold $\frac{p}{100}$. The classifier is considered as totally fair when this ratio satisfies a 100%-rule. Conversely, a 0%-rule indicates a completely unfair model.

$$\textit{P-rule:}\ \min\left(\frac{P(\widehat{Y}=1|S=1)}{P(\widehat{Y}=1|S=0)}, \frac{P(\widehat{Y}=1|S=0)}{P(\widehat{Y}=1|S=1)}\right) \quad (1)$$

An algorithm is considered fair if across both demographics $S=0$ and $S=1$, for the outcome $Y=1$ the predictor $\widehat{Y}$ has equal *true* positive rates, and for $Y=0$ the predictor $\widehat{Y}$ has equal *false* positive rates [3]. This constraint enforces that accuracy is equally high in all demographics since the rate of positive and negative classification is equal across the groups.

Definition 2. $P(\widehat{Y}=1|S=0, Y=y) = P(\widehat{Y}=1|S=1, Y=y), \forall y \in \{0,1\}$

A metric to assess this objective is to measure the disparate mistreatment (DM) [4]. It computes the absolute difference between the false positive rate (FPR) and the false negative rate (FNR) for both demographics.

$$D_{FPR} : |P(\widehat{Y}=1|Y=0,S=1) - P(\widehat{Y}=1|Y=0,S=0)| \quad (2)$$

$$D_{FNR} : |P(\widehat{Y}=0|Y=1,S=1) - P(\widehat{Y}=0|Y=1,S=0)| \quad (3)$$

The closer the values of D_{FPR} and D_{FNR} to 0, the lower the degree of disparate mistreatment of the classifier.

B. Related Work

Recent research in fair machine learning has made considerable progress in quantifying and mitigating undesired bias. Three different types of mitigation strategies exist: The family of "pre-processing" algorithms which ensures that the input data is fair, "in-processing" methods where the undesired bias is directly mitigated during the training phase, and finally "post-processing" algorithms where the output of the trained classifier is modified.

We propose an "in-processing" algorithm. Here, undesired bias is directly mitigated during the training phase. A straightforward approach to achieve this goal is to integrate a fairness penalty directly in the loss function. One such algorithm integrates a decision boundary covariance constraint for logistic regression or linear SVM [5]. In another approach, a meta algorithm takes the fairness metric as part of the input and returns a new classifier optimized towards that fairness metric [6]. Furthermore, the emergence of generative adversarial networks (GANs) provided the required underpinning for fair classification using adversarial debiasing [7]. In this field, a neural network classifier is trained to predict the label Y, while simultaneously minimizing the ability of an adversarial neural network to predict the sensitive attribute S [8, 9, 10].

III. Fair Adversarial Gradient Tree Boosting (FAGTB)

Our aim is to learn a classifier that is both effective for predicting true labels and fair, in the sense that it cares about metrics defined in section II-A for demographic parity or equalized odds. The idea is to leverage the great performance of gradient tree boosting (GTB) for classification, while adapting it for fair machine learning via adversarial learning.

GTB processes sequentially by gradient iteration to define a prediction function of the form:

$$F_M(x_i) = \sum_{m=0}^{M} \gamma_m h_m(x_i) \quad (4)$$

where x_i is a predictor vector, M is the total number of iterations, and $h_m(x_i)$ corresponds to a weak learner at step m in the form of a greedy CART prediction. Given a loss function $\mathcal{L}(y_i, F(x_i))$ to minimize for all (x_i, y_i) from the training set, GTB calculates at each step m the so-called "pseudo residuals":

$$r_{im} = -\left[\frac{\partial \mathcal{L}(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n \quad (5)$$

Then, at each step, GTB fits a new weak learner $h_m(x)$ to those pseudo residuals and adds it to the current model. This step is repeated until the algorithm converges.

This architecture allows us to apply for fair classification with decision tree algorithms the concept of adversarial learning, which corresponds to a two-player game with two contradictory components, such as in generative adversarial networks (GAN) [7].

A. Min-Max formulation

In the vein of [8, 10, 9] for fair classification, we consider a predictor function F, that outputs the probability of an input vector X for being labelled $Y=1$, and an adversarial model A which tries to predict the sensitive attribute S from the output of F. Depending on the accuracy rate of the adversarial algorithm, we penalize the gradient of the GTB at each iteration. The goal is to obtain a classifier F whose outputs do not allow the adversarial function to reconstruct the value of the sensitive attribute. If this objective is achieved, the data bias in favor of some demographics disappeared from the output prediction.

The predictor and the adversarial classifiers are optimized simultaneously in a min-max game defined as:

$$\arg\min_{F} \max_{\theta_A} \sum_{i=1}^{n} \mathcal{L}_{F_i}(F(x_i)) - \lambda \sum_{i=1}^{n} \mathcal{L}_{A_i}(F(x_i);\theta_A) \quad (6)$$

where $\mathcal{L}_{F_i}$ and $\mathcal{L}_{A_i}$ are respectively the predictor and the adversary loss for the training sample i given $F(x_i) \in \mathbb{R}$, which refers to the output of the GTB predictor for input x_i. The hyperparameter λ controls the impact of the adversarial loss.

The targeted classifier outputs the label $\widehat{Y}$ which maximizes the posterior $P(\widehat{Y}|X)$. Thus, for a given sample x_i, we get:

$$\hat{y}_i = \arg\max_{y \in \{0;1\}} p_F(Y=y|X=x_i) \quad (7)$$

where $p_F(Y=1|X=x_i) = \sigma(F(x_i))$, with σ denoting the sigmoid function. Therefore, $\mathcal{L}_{F_i}$ is defined as the negative log-likelihood of the predictor for the training sample i:

$$\begin{aligned}\mathcal{L}_{F_i}(F(x_i)) &= -\log p_F(Y=y_i|X=x_i) \\ &= -\mathbf{1}_{y_i=1}\log(\sigma(F(x_i))) \\ &\quad - \mathbf{1}_{y_i=0}\log(1-\sigma(F(x_i)))\end{aligned} \quad (8)$$

where $\mathbf{1}_{cond}$ equals 1 if $cond$ is true, 0 otherwise.

The adversary A corresponds to a neural network with parameters θ_A, which takes as input the sigmoid of the predictor's output for any sample i (i.e., $P_F(Y=1|X=x_i)$), and outputs the probability P_{F,θ_A} for the sensitive to equal 1:

- For the demographic parity task, $P_F(Y=1|X=x_i)$ is the only input given to the adversary for the prediction of the sensitive attribute s_i. In that case, the network A outputs the conditional probability $P_{F,\theta_A}(S=1|V=v_i) = A(v_i)$, with $V = (\sigma(F(X)))$.
- For the equalized odds task, the label y_i is concatenated to $P_F(Y=1|X=x_i)$ to form the input vector of the

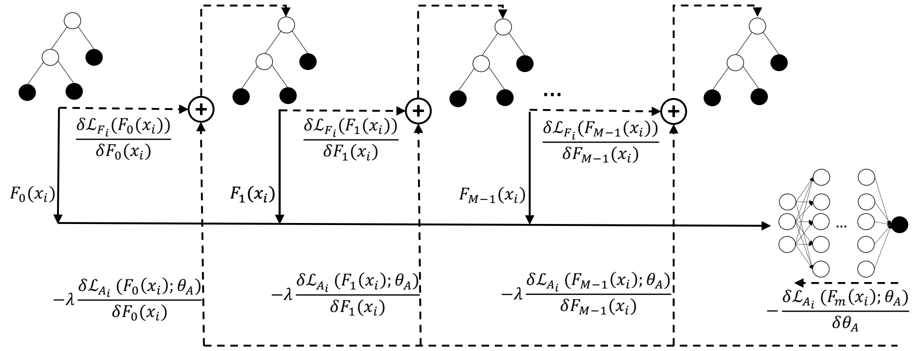

Fig. 1: The architecture of the Fair Adversarial Gradient Tree Boosting (FAGTB). Four steps are depicted, each one corresponding to a tree h that is added to the global classifier F. The neural network on the right is the adversary that tries to predict the sensitive attributes from the outputs of the classifier. Solid lines represents forward operations, while dashed ones represent gradient propagation. At each step m, gradients from the prediction loss and the adversary loss are summed to form the target for the next decision tree h_{m+1}.

adversary $v_i = (\sigma(F(x_i)), y_i)$, so that the function A could be able to output different conditional probabilities $P_{F,\theta_A}(S = 1|V = v_i)$ depending on the label y_i of i.

The adversary loss is defined for any training sample i as:

$$\begin{aligned} \mathcal{L}_{A_i}(F(x_i);\theta_A) = &- \mathbf{1}_{s_i=1} \log(\sigma(A(v_i))) \\ &- \mathbf{1}_{s_i=0} \log(1-\sigma(A(v_i))) \end{aligned} \tag{9}$$

with v_i defined according to the task as detailed above.

Note that, for the case of demographic parity, if there exists (F^*, θ_A^*) such that $\theta_A^* = \arg\max_{\theta_A} P_{F^*,\theta_A}(S|V)$ on the training set, $P_{F^*,\theta_A^*}(S|V) = \widehat{P}(S)$ and $P_{F^*}(Y|X) = \widehat{P}(Y|X)$, with $\widehat{P}(S)$ and $\widehat{P}(Y|X)$ the corresponding distributions on the training set, (F^*, θ_A^*) is a global optimum of our min-max problem eq. (6). In that case, we have both a perfect classifier in training, and a completely fair model since the best possible adversary is not able to predict S more accurately than the estimated prior distribution. Similar observations can easily be made for the equalized odds task (by replacing $\widehat{P}(S)$ by $\widehat{P}(S|Y)$ and using the corresponding definition of V in the previous assertion). While such a perfect setting does not always exists in the data, it shows that the model is able to identify a solution when it reaches one. If a perfect solution does not exists in the data, the optimum of our min-max problem is a trade-off between prediction accuracy and fairness, controlled by the hyperparameter λ.

B. Learning

The learning process is outlined as pseudo code in Algorithm 1. The algorithm first initializes the classifier F_0 with constant values for all inputs, as done for the classical GBT. Additionally, it initializes the parameters θ_A of the adversarial neural network A (a Xavier initialization is used in our experiments). Then, at each iteration m, beyond calculating the pseudo residuals r_{im} for any training sample i w.r.t. the targeted prediction loss $\mathcal{L}_{F_i}$, it computes pseudo residuals t_{im} for the adversarial loss $\mathcal{L}_{A_i}$ too. Both residuals are combined in $u_{im} = r_{im} - \lambda * t_{im}$, where λ controls the impact of the adversarial network. The algorithm then fits a new weak regressor h_m (a decision tree in our work) to residuals using the training set $\{(x_i, u_{im})\}_{i=1}^n$. This pseudo-residuals regressor is supposed to correct both prediction and adversarial biases of the old classifier F_{m-1}. It is added to it after a line search step, which determines the best γ_m weight to assign to h_m in the new classifier F_m. Finally, the adversarial has to adapt its weights according to new outputs (i.e., using the training set $\{(F_m(x_i), s_i)\}_{i=1}^n$). This is done by gradient backpropagation. A schematic representation of our approach can be found in Figure 1.

IV. Experiments

For our experiments we use four different popular data sets often used in fair classification: The Adult UCI Income data set [11], the COMPAS data set [12], the Default data set [13], the Bank Marketing data set [14].

For all data sets, we repeat 10 experiments by randomly sampling two subsets, 80% for the training set and 20% for the test set. Finally, we report the average of the accuracy and the fairness metrics from the test set.

Because different optimization objectives result in different algorithms, we run separate experiments for the two fairness metrics of our interest, demographic parity (Table I) and equalized odds (Table II). More specifically, for demographic parity we aim at a p-rule of 90% for all algorithms and then compare the accuracy. Optimizing for equalized odds, results are more difficult to compare. In order to be able to compare the accuracy, we have done our best to obtain, each time, a disparate level below 0.03.

As a baseline, we use a classical, "unfair" gradient tree boosting algorithm, Standard GTB, and a deep neural network, Standard NN.

Further, to evaluate if the complexity of the adversarial network has an impact on the quality of the results, we compare a simple logistic regression adversarial, FAGTB-1-Unit, with a complex deep neural network, FAGTB-NN.

Algorithm 1 Fair Adversarial Gradient Tree Boosting

Input: training set $(x_i, s_i, y_i)_{i=1}^n$, a number of iterations M, an adversarial learning rate α, a differentiable loss function $\mathcal{L}_F$ for the output classifier and $\mathcal{L}_A$ for the adversarial classifier.

Initialize: Calculate the constant value:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} \mathcal{L}_{F_i}(\gamma)$$

Initialize parameters θ_A of the neural network $A(x)$

for $m = 1$ **to** $M - 1$ **do**

(a) Calculate the pseudo residuals:

$$r_{im} = -\left[\frac{\partial \mathcal{L}_{F_i}(F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n$$

(b) Calculate the pseudo residuals of the adversarial from the input $F_{m-1}(x_i)$:

$$t_{im} = -\left[\frac{\partial \mathcal{L}_{A_i}(F(x_i;\theta_A))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n$$

(c) Calculate the training loss derivative:

$$u_{im} = r_{im} - \lambda * t_{im}$$

(d) Fit a classifier $h_m(x)$ to pseudo residuals using the training set $\{(x_i, u_{im})\}_{i=1}^n$

(e) Compute multiplier γ_m by solving the following one-dimensional optimization problem:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} \mathcal{L}_{F_i}(F_{m-1}(x_i) + \gamma * h_m(x_i)) - \lambda * \mathcal{L}_{A_i}(F_{m-1}(x_i) + \gamma * h_m(x_i); \theta_A).$$

(f) Update the learning model:

$$F_m(x_i) = F_{m-1}(x_i) + \gamma_m * h_m(x_i)$$

(g) Fit the adversarial A to the using the new outputs (i.e., using the training set $\{(F_m(x_i), s_i)\}_{i=1}^n$)

$$\theta_A := \theta_A - \alpha * \frac{\partial \mathcal{L}_{A_i}(F_m(x_i);\theta_A)}{\partial \theta_A}$$

end do

In addition to the algorithms mentioned above, we evaluate the following fair state-of-the-art in-processing algorithms: Wadsworth2018 [9][3], Zhang2018 [8][4], Kamishima [15][2] Feldman [16][2], Zafar-DI [17][2] and Zafar-DM [4][2].

[2]https://github.com/algofairness/fairness-comparison
[3]https://github.com/equialgo/fairness-in-ml
[4]https://github.com/IBM/AIF360

For each algorithm and for each data set, we obtain the best hyperparameters by grid search in 5-fold cross validation (specific to each of them). As a reminder, for FAGTB the λ value is used to balance the 2 cost functions during the training phase. This value depends exclusively on the main objective: For example, to obtain the demographic parity objective with 90% p-rule, we choose a lower and thus less weighty λ than for a 100% p-rule objective. In order to better understand this hyperparameter λ we illustrate its impact on the accuracy and the p-rule metric in Figure 2 for the Adult UCI data set. For that, we model the FAGTB-NN algorithm with 10 different values of λ and we run each experiment 10 times. In the graph, we report the accuracy and the p-rule fairness metric, and finally plot a polynomial regression of second order to demonstrate the general effect.

For Standard GTB, we parameterize the number of trees and the maximum tree depth. For example, for the Bank data set, a tree depth of 3 with 800 trees is sufficient. For the Standard NN, we parameterize the number of hidden layers and units with a ReLU function and we apply a specific dropout regularization to avoid overfitting. Further, we use an Adam optimisation with a binary cross entropy loss. For the Adult UCI data set for example, the architecture consists of 2 hidden layers with 16 and 8 units, respectively, and ReLU activations. The output layer comprises one single output node with sigmoid activation.

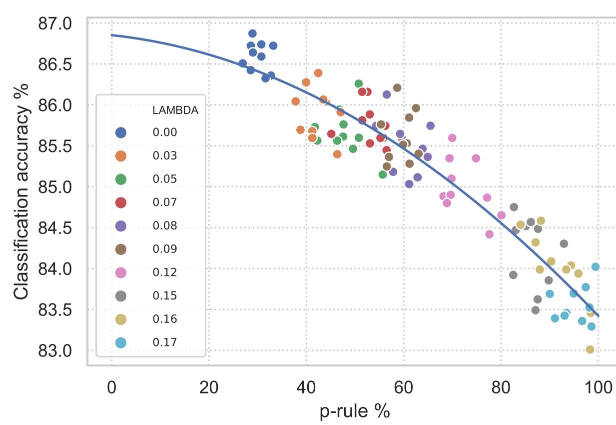

Fig. 2: Impact of hyperparameter λ (Adult UCI data set): Higher values of λ produce fairer predictions, while λ near 0 allows to only focus on optimizing the classifier predictor.

For FAGTB, to accelerate the learning phase, we decided to sacrifice some performance by replacing the one-dimensional optimization γ_m by a specific fixed learning rate for the classifier predictor. All hyperparameters mentioned above, for trees and neural networks, are selected jointly. Notice that those choices impact the rapidity of convergence for each of them. For example, if the classifier predictor converges too quickly this may result in biased prediction probabilities during the first iterations which are difficult to correct by the adversary afterwards. For FAGTB-NN, in order to achieve better results, we execute for each gradient boosting iteration

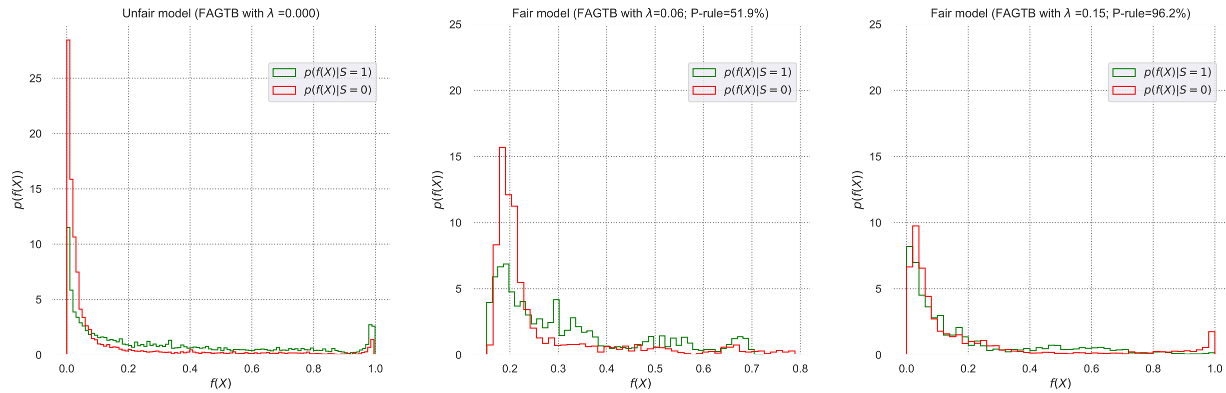

Fig. 3: Distributions of the predicted probabilities given the sensitive attribute S (Adult UCI data set)

several training iterations of the adversarial NN. This produces a more persistent adversarial algorithm. Otherwise, the predictor classifier GTB could dominate the adversary. At the first iteration, we begin with modeling a biased GTB and we then model the adversarial NN based on those biased predictions. This approach allows to have a better weight initialization of the adversarial NN. It is more suitable for the specific bias on the data set. Without this specific initialization we encountered some cases where the predictor classifier surpasses the adversarial too quickly and tends to dominate from the beginning. Compared to the FAGTB-NN, the adversary of the FAGTB-1-Unit is more simple. In this case, the two parameters of the adversarial are chosen randomly and for each gradient boosting iteration only one is computed for the adversarial unit.

For demographic parity (Table I), as expected Standard GTB and Standard NN achieve the highest accuracy. However, they are also the most biased ones. For example, the classical gradient tree boosting algorithm achieves a 32.6% p-rule for the Adult UCI data set. In this particular case, the prediction for earning a salary above $50,000 is in average more than three times higher for men than for women. Comparing the mitigation algorithms, FAGTB-NN achieves the best result with the highest accuracy while maintaining a reasonable high p-rule equality (90%). The choice of a neural network architecture for the adversary proved to be in any case better than a simple logistic regression. This is particularly true for the COMPAS data set where, for a similar p-rule, the difference in accuracy is considerable (2.7 points). Recall that for demographic parity the adversarial classifier only has one single input feature which is the output of the prediction classifier. It seems necessary to be able to segment this input in several ways to better capture information relevant to predict the sensitive attribute. The sacrifice of accuracy is less important for the Bank and the Default data set. The dependence between the sensitive attribute and the target label is thus less important than for the COMPAS data set. To achieve a p-rule of 90%, we sacrifice 4.6 points of accuracy (Comparing GTB and FAGTB-NN) for COMPAS, 0.7 points for Default and 0.6 points for Bank.

In Figure 3 we plot the distribution of the predicted probabilities for each sensitive attribute S for 3 different models: An unfair model with $\lambda = 0$, and 2 fair FAGTB models with $\lambda = 0.06$ and $\lambda = 0.15$, respectively. For the unfair model, the distribution differs most for the lower probabilities. The second graph shows an improvement but there remain some differences. For the final one, the distributions are practically aligned.

Zhang2018 [8] introduced a projection term which ensures that the predictor never moves in a direction that could help the adversary. While this is an interesting approach, we noticed that this term does not improve the results for demographic parity. In fact, the Wadsworth2018 [9] algorithm follows the same approach but without projection term and obtains similar results.

For equalized odds, the min-max optimization is more difficult to achieve than demographic parity. The fairness metrics D_{FPR} and D_{FNR} are not exactly comparable thus we did not succeed to obtain the same level of fairness. However, we notice that the FAGTB-NN achieves better accuracy with a reasonable level of fairness. Concretely, we achieve for the 4 data sets and for both metrics values below 0.02 or less, except for the Bank data set where D_{FNR} is equal to 0.07. For this data set, most of the state-of-the-art algorithms result in a D_{FNR} between 0.06 and 0.08. The reason why it proves hard to achieve a low False Negative Rate (FNR), is that the total share of the target is very low at 11.7%. A possible way to handle this problem of imbalanced target class could be to to add a specific weight directly in the loss function. We also notice that the difference in the results between FAGTB-1-Unit and FAGTB-NN is much more significant, one possible reason is that an unique logistic regression cannot keep a sufficient amount of information in order to predict the sensitive attribute.

V. Conclusion

In this work, we developed a new approach to produce fair gradient boosting algorithms. Compared with other state-of-the-art algorithms, our method proves to be more efficient in terms of accuracy while obtaining a similar level of fairness.

TABLE I: Results for Demographic Parity

	Adult		COMPAS		Default		Bank	
	Acc.	P-rule	Acc.	P-rule	Acc.	P-rule	Acc.	P-rule
Standard GTB	86.8%	32.6%	69.1%	61.2%	82.9%	77.2%	90.8%	48.1%
Standard NN	85.3%	31.4%	67.5%	71.1%	82.1%	63.3%	90.3%	58.6%
FAGTB-1-Unit	84.4%	90.4%	61.8%	90.1%	81.5%	90.1%	90.1%	90.0%
FAGTB-NN	84.9%	90.3%	64.5%	90.0%	82.2%	90.2%	90.2%	90.0%
Wadsworth2018 [9]	83.1%	89.7%	63.9%	90.1%	81.8%	90.0%	90.2%	90.1%
Zhang2018 [8]	83.3%	90.0%	64.1%	89.8%	81.4%	90.0%	90.0%	90.0%
Zafar-DI [5]	82.2%	89.8%	63.9%	89.7%	80.7%	89.8%	89.2%	90.1%
Kamishima [15]	82.3%	89.9%	63.8%	90.0%	81.1%	90.0%	89.6%	89.9%
Feldman [16]	-	-	61.4%	90.1%	72.2%	90.2%	-	-

Comparing our approach with different common fair algorithms by accuracy and fairness (p-rule metric) for the Adult UCI, the COMPAS, the Default and the Bank data set.

TABLE II: Results for Equalized Odds

	Adult			COMPAS			Default			Bank		
	Acc.	D_{FPR}	D_{FNR}	Acc.	D_{FPR}	D_{FNR}	Acc.	D_{FPR}	D_{FNR}	Acc.	D_{FPR}	D_{FNR}
Standard GTB	86.8%	0.06	0.07	69.1%	0.12	0.20	82.9%	0.02	0.04	90.8%	0.04	0.06
Standard NN	85.3%	0.07	0.10	67.5%	0.09	0.15	82.1%	0.02	0.05	90.3%	0.05	0.08
FAGTB-1-Unit	86.3%	0.02	0.02	65.1%	0.03	0.12	82.1%	0.00	0.01	89.7%	0.02	0.07
FAGTB-NN	86.4%	0.02	0.02	66.2%	0.01	0.02	82.5%	0.00	0.01	90.3%	0.01	0.07
Wadsworth2018 [9]	84.9%	0.02	0.03	65.4%	0.02	0.03	81.2%	0.01	0.02	89.4%	0.01	0.07
Zhang2018 [8]	84.8%	0.03	0.03	64.9%	0.03	0.02	81.9%	0.00	0.01	89.8%	0.00	0.07
Zafar-DM [4]	83.9%	0.03	0.09	64.3%	0.09	0.17	81.0%	0.00	0.03	89.5%	0.01	0.08
Kamishima [15]	82.6%	0.06	0.24	63.6%	0.08	0.11	80.5%	0.00	0.04	89.3%	0.00	0.08
Feldman [16]	80.6%	0.07	0.05	61.1%	0.03	0.03	71.8%	0.02	0.02	87.1%	0.05	0.06

Comparing our approach with different common fair algorithms by accuracy and fairness (D_{FPR}, D_{FNR}) for the Adult UCI, the COMPAS, the Default and the Bank data set.

References

[1] C. Zhang, C. Liu, X. Zhang, and G. Almpanidis, "An up-to-date comparison of state-of-the-art classification algorithms," *Expert Systems with Applications*, vol. 82, 04 2017.

[2] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *ITCS'12*. ACM, 2012, pp. 214–226.

[3] M. Hardt, E. Price, ecprice, and N. Srebro, "Equality of opportunity in supervised learning," in *NIPS'16*, 2016, pp. 3315–3323.

[4] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment," in *WWW'17*, 2017, pp. 1171–1180.

[5] M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi, "Fairness Constraints: Mechanisms for Fair Classification," in *AISTATS'17*, A. Singh and J. Zhu, Eds., vol. 54, 2017, pp. 962–970.

[6] L. E. Celis, L. Huang, V. Keswani, and N. K. Vishnoi, "Classification with fairness constraints: A meta-algorithm with provable guarantees," *CoRR*, vol. abs/1806.06055, 2018.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS'14*, 2014, pp. 2672–2680.

[8] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *AIES '18*. ACM, 2018, pp. 335–340.

[9] C. Wadsworth, F. Vera, and C. Piech, "Achieving fairness through adversarial learning: an application to recidivism prediction," *CoRR*, vol. abs/1807.00199, 2018.

[10] G. Louppe, M. Kagan, and K. Cranmer, "Learning to pivot with adversarial networks," in *NIPS'17*, 2017, pp. 982–991.

[11] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[12] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine bias. ProPublica, May 23, 2016," 2016.

[13] I.-C. Yeh and C.-h. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 2473–2480, Mar. 2009.

[14] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decision Support Systems*, vol. 62, 06 2014.

[15] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *ECML/PKDD'12*, 2012, pp. 35–50.

[16] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *KDD '15*, 2015, pp. 259–268.

[17] M. B. Zafar, I. Valera, M. Rodriguez, K. Gummadi, and A. Weller, "From parity to preference-based notions of fairness in classification," in *NIPS'17*, 2017, pp. 229–239.