Data Science and Engineering (2020) 5:99–110
https://doi.org/10.1007/s41019-020-00124-2

Achieving Fairness with Decision Trees: An Adversarial Approach

Vincent Grari[1] · **Boris Ruf**[2] · **Sylvain Lamprier**[1] · **Marcin Detyniecki**[2]

Received: 18 February 2020 / Revised: 28 April 2020 / Accepted: 5 May 2020 / Published online: 20 May 2020

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 (Zhang et al. in Expert Syst Appl 82:128–150, 2017). 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 four 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.

Keywords Fair machine learning · Adversarial · Gradient boosting

1 Introduction

Machine learning models are increasingly used in decision making processes. In many fields of application, they generally deliver superior performance compared with conventional, deterministic algorithms. However, those models are mostly black boxes which are hard, if not impossible, to interpret. 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. In particular, algorithms trained on biased data are prone to learn, perpetuate or even reinforce these biases [2]. In recent years, many incidents of this nature have been documented. For example, an algorithmic model used to generate predictions of criminal recidivism in the USA (COMPAS) discriminated against black defendants [3]. Also, discrimination based on gender and race could be demonstrated for targeted and automated online advertising on employment opportunities [4]. In this context, the EU introduced the General Data Protection Regulation (GDPR) in May 2018. This legislation represents one of the most important changes in the regulation of data privacy in more than 20 years. It strictly regulates the collection and use of sensitive personal data. With the aim of obtaining non-discriminatory algorithms, it rules in Article 9(1): "Processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation shall be prohibited" [5]. One fairness method often used in practice today is to remove protected attributes from the data set. This concept is known as "fairness through unawareness" [6]. While this approach may prove viable when using conventional, deterministic algorithms with a manageable quantity of data, it is insufficient for machine learning algorithms trained on "big data." Here, complex correlations in the data may provide unexpected links to sensitive information. This way, presumably non-sensitive attributes, can serve as substitutes or proxies for protected attributes.

✉ Vincent Grari
vincent.grari@lip6.fr

Boris Ruf
boris.ruf@axa.com

Sylvain Lamprier
sylvain.lamprier@lip6.fr

Marcin Detyniecki
marcin.detyniecki@axa.com

[1] LIP6/CNRS, Sorbonne Université, Paris, France

[2] AXA REV Research, Paris, France

For this reason, next to optimizing the performance of a machine learning model, the new challenge for data scientists is to determine whether the model output predictions are discriminatory, and how they can mitigate such unwanted bias as much as possible.

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 threefold:

- To the best of our knowledge, we propose the first adversarial learning method for generic classifiers, including non-differentiable machines, such as decision trees;
- 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.

The remainder of this paper proceeds as follows: First, Sect. 2.1 presents our notation and introduces common definitions of fairness which will serve as metrics to measure the performance of our approach. Then, Sect. 2.2 reviews papers related with our work. Section 3 briefly recaps the principle of classical gradient tree boosting. Next, Sect. 4 outlines a novel algorithm which combines gradient tree boosting with adversarial debiasing. Finally, Sect. 5 presents experimental results of our approach.

2 Fair Machine Learning

2.1 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 ith example, s_i is its binary sensitive attribute and y_i is 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.

2.1.1 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 [7]. The underlying idea is that each demographic group has the same chance for a positive outcome.

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:

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

The second metric available for demographic parity is the disparate impact (DI) assessment [8]. It considers the absolute difference of outcome distributions for subpopulations with different sensitive attribute values. The smaller the difference, the fairer the model:

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

2.1.2 Equalized Odds

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 [9]. This constraint enforces that accuracy is equally high in all demographics since the rate of positive and negative classification is equal across the groups. The notion of fairness here is that chances of being correctly or incorrectly classified positive should be equal for every group.

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) [10]. It computes the absolute

difference between the false positive rate (FPR) and the false negative rate (FNR) for both demographics:

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

$$D_{FNR} : |P(\widehat{Y} = 0|Y = 1, S = 1) - P(\widehat{Y} = 0|Y = 1, S = 0)| \tag{4}$$

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

2.2 Related Work

Recently, research in fair machine learning has prospered, and considerable progress was made when it comes to quantifying and mitigating undesired bias. For the mitigation strategies, three distinct approaches exist.

Algorithms which belong to the "pre-processing" family ensure that the input data are fair. This can be achieved by suppressing the sensitive attributes, by changing class labels of the data set and by reweighting or resampling the data [11–13].

The second type of mitigation strategies comprises the "in-processing" algorithms. 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 [14]. In another approach, a meta-algorithm takes the fairness metric as part of the input and returns a new classifier optimized toward that fairness metric [15]. Furthermore, the emergence of generative adversarial networks (GANs) provided the required underpinning for fair classification using adversarial debiasing [16]. 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 [17–19].

The final group of mitigation algorithms follows a post-processing" approach. In this case, only the output of a trained classifier is modified. A Bayes optimal equalized odds predictor can be used to change output labels with respect to an equalized odds objective [9]. A different paper presents a weighted estimator for demographic disparity which uses soft classification based on proxy model outputs [20]. The advantage of post-processing algorithms is that fair classifiers are derived without the necessity of retraining the original model which may be time-consuming or difficult to implement in production environments. However, this approach may have a negative effect on accuracy or could compromise any generalization acquired by the original classifier [21].

Algorithm 1 Classical Gradient Boosting

Input: Training set $(x_i, s_i, y_i)_{i=1}^n$, a number of iterations M, a differentiable loss function $\mathcal{L}(y, F(x))$

Initialize: Calculate the constant value:

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

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

(a) Calculate the pseudo residuals:

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

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

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

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} \mathcal{L}\left(y_i, F_{m-1}(x_i) + \gamma * h_m(x_i)\right)$$

(d) Update the model:

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

end for

3 Gradient Tree Boosting

In order to establish the basis for our approach and also to introduce our notation, we first summarize the principle of classical gradient tree boosting. The "gradient boosting machine" (GBM) constitutes a prediction model for regression and classification problems based on an ensemble technique where multiple weak learners are combined to produce a strong learner [22]. Often, such weak learners are decision trees, generally of the type classification and regression tree (CART). In this case, the algorithm is called gradient tree boosting (GTB). The weak learners are built sequentially. Eventually, a strong classifier is obtained as a weighted sum of the weak learners. The classical gradient descent algorithm is used to optimize the model by any differentiable loss function.

The objective of the GBM is to find a good estimate of the function F which approximately minimizes the empirical loss function:

$$\min_{F} \sum_{i=1}^{n} \mathcal{L}(y_i, F(x_i)) \tag{5}$$

where the loss function $\mathcal{L}(y_i, F(x_i))$ measures the ith prediction compared to the true label. In the classical version of the GBM, the prediction corresponding to a feature vector x is given by an additive model of the form:

$$F_M(x_i) = \sum_{m=0}^{M} \gamma_m h_m(x_i) \tag{6}$$

where M is the total number of iterations, and $h_m(x_i)$ corresponds to a weak learner at step m (a greedy CART predictor in the following).

The main steps for fitting the model are shown as pseudocode in Algorithm 1. The method exploits the fact that the residual corresponds to the negative gradient of the loss function. Thus, we calculate at each step m the so-called pseudoresiduals:

$$r_{im} = -\left[\frac{\partial \mathcal{L}(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n \tag{7}$$

In order to update the model, we fit a new weak learner $h_m(x)$ to those pseudoresiduals and add it to the current model. This step is repeated until the algorithm converges.

4 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 Sect. 2.1 for demographic parity or equalized odds. The idea is to leverage the great performance of GTB for classification, while adapting it for fair machine learning via adversarial learning.

4.1 Min–Max Formulation

While most state-of-the-art algorithms focus on the independence of the predicted probability predictions.

The GTB processes sequentially by gradient iteration (Sect. 3). 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 network (GAN) [23]. In the vein of [17–19] 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:

$$\arg\min_{F} \max_{\theta_A} \sum_{i=1}^{n} \mathcal{L}_{F_i}(F(x_i)) - \lambda \sum_{i=1}^{n} \mathcal{L}_{A_i}(F(x_i); \theta_A) \tag{8}$$

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:

$$\hat{y}_i = \arg\max_{y \in \{0;1\}} p_F(Y = y | X = x_i) \tag{9}$$

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:

$$\begin{aligned} \mathcal{L}_{F_i}(F(x_i)) &= -\log p_F(Y = y_i | X = x_i) \\ &= -\mathbf{1}_{y_i=1} \log(\sigma(F(x_i))) \\ &\quad - \mathbf{1}_{y_i=0} \log(1 - \sigma(F(x_i))) \end{aligned} \tag{10}$$

where $\mathbf{1}_{cond}$ equals 1 if $cond$ is true and 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 equal to 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 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 then defined for any training sample i as:

$$\begin{aligned} \mathcal{L}_{A_i}(F(x_i); \theta_A) &= -\mathbf{1}_{s_i=1} \log(\sigma(A(v_i))) \\ &\quad - \mathbf{1}_{s_i=0} \log(1 - \sigma(A(v_i))) \end{aligned} \tag{11}$$

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 $\widehat{P}(Y|X)$ being the corresponding distributions on the training set, and (F^*, θ_A^*) is a global optimum of our min–max problem Eq. (8). 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 exist in the data, it shows that the model is able to identify a solution when it reaches one. If a perfect solution does not exist in the data, the optimum of our min–max problem is a trade-off between prediction accuracy and fairness, controlled by the hyperparameter λ.

Algorithm 2 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:

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

(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

4.2 Learning

The learning process is outlined as pseudocode in Algorithm 2. 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 pseudoresiduals r_{im} for any training sample i w.r.t. the targeted prediction loss $\mathcal{L}_{F_i}$, it computes pseudoresiduals 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 pseudoresiduals 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 is shown in Fig. 1.

5 Empirical Results

We evaluate the performance of our algorithm empirically with respect to regression accuracy and fairness. We conduct the experiments on a synthetic scenario, but also on real-world data sets. Finally, we compare the results with state-of-the-art algorithms.

5.1 Synthetic Scenario

We illustrate the fundamental functionality of our proposal with a simple toy scenario which was inspired by the Red Car example [24]. The subject is a pricing algorithm for a fictional car insurance policy. The purpose of this exercise is to train a fair classifier which estimates the claim likelihood without incorporating any gender bias. We want to demonstrate the effects of an unfair model versus a fair model.

We focus on the general claim likelihood and ignore the severity or cost of the claim. Further, we only consider the binary case of claim or not (as opposed to a frequency). We assume that the claim likelihood only depends on the aggressiveness and the inattention of the policyholder. To make the training more complex, these two properties are not directly represented in the input data but only

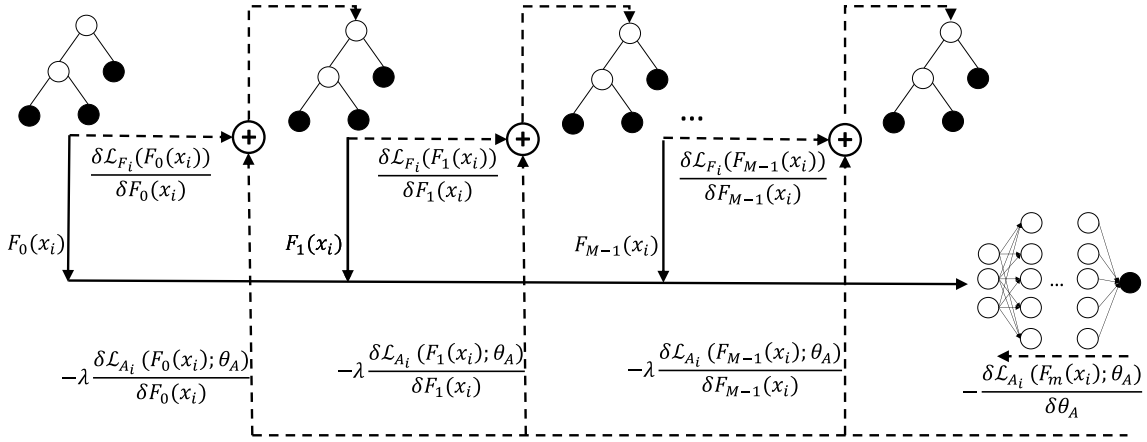

Fig. 1 The architecture of the fair adversarial gradient tree boosting (FAGTB). Four steps are depicted, each one corresponding to a tree h that is added to the global classifier F. The neural network on the right is the adversary that tries to predict the sensitive attributes from the outputs of the classifier. Solid lines represent 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}

indirectly available through correlations with other input features. We create a binary label Y with no dependence with the sensitive attribute S. Concretely, we use as features the protected attribute *gender* of the policyholder, the unprotected attributes *color* of the car and *age* of the policyholder. In our data distribution, the *color* of the car is strongly correlated with both *gender* and aggressiveness. The *age* is not correlated with *gender*. However, the *age* is correlated with the inattention of the policyholder. Thus, the latter input feature is actually linked to the claim likelihood.

First, we generate the training samples $(x_i, s_i, y_i)_{i=1}^n$. The unprotected attributes $x_i = (c_i, a_i)$ represent the *color* of the car and the *age* of the policyholder, respectively. s is the protected variable *gender*. y is the binary class label, where $y = 1$ indicates a registered claim. As stated above, we do not use the two features aggressiveness (A) and inattention (I) as input features but only to construct the data distribution which reflects the claim likelihood. In order to make it more complex, we add a little noise ϵ_i. These training samples are generated as follows: for each i, let s be a discrete variable with the discrete uniform distribution such that $s_i \in [0, 1]$:

$$\begin{pmatrix} I_i \\ a_i \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 40 \end{pmatrix}, \begin{pmatrix} 1 & 4 \\ 4 & 20 \end{pmatrix}\right]$$
$$A_i \sim \mathcal{N}(0, 1)$$
$$c_i = (1.5 * s_i + A_i) > 1$$
$$y_i = \sigma(A_i + I_i + \epsilon_i) > 0.5$$
$$\epsilon_i \sim \mathcal{N}(0, 0.1)$$

A correlation matrix of the distribution is shown in Table 1.

Table 1 Correlation matrix of the synthetic scenario

a	1.0				
A	0.01	1.0			
c	− 0.01	0.68	1.0		
s	0.0	− 0.01	0.36	1.0	
I	0.90	0.01	0.0	0.0	1.0
	a	A	c	s	I

The features are: age (a), aggressivity (A), color (c), gender (s), inattention (I)

We execute first a classical GTB algorithm. In Fig. 2, first graph, we can see the curves of accuracy and the fairness metric p-rule during the training phase. The model shows a stability of the two objectives, this being due to the lack of information and the small number of explanatory variables. Even though there is no obvious link with the sensitive attribute, we notice that this model is unfair (p-rule of 67%). In fact, the outcome observations Y depend exclusively on A and I which should have no dependence with the sensitive feature S. To reconstruct the aggressiveness, the classifier has to consider the color of the car. Unfortunately, it incorporates the sensitive information too, resulting in a claim likelihood prediction one and a half times more for men than for women (1/0.67).

To solve this problem and, thus, to achieve demographic parity, we use the FAGTB algorithm with a specific hyperparameter λ. This hyperparameter is obtained by tenfold cross-validation on 20% of the test set. As explained above, the choice of this value depends on the main objective, resulting in a trade-off between accuracy and fairness. We decided to train a model that reaches a p-rule of approximately 95% with a λ equal to 0.015.

In Fig. 2, we also plot five other models with different values of λ, optimized for demographic parity. We observe that during training, when the attenuation of the bias is sudden, the accuracy also dramatically drops. We note that gaining 29 points of p-rule leads to a decrease in accuracy of ten points. To have a better understanding of what is happening when we consider the model as fair in this specific scenario, we plot the features importance permutation for the fair and the unfair model in Fig. 3. The model reported importance on the age feature, which is not correlated with the sensitive. The difference between the two features is higher for the fair model (0.145 points), the color feature becoming insignificant. With higher lambda values, the weight of this indirectly correlated feature would tend to 0.

5.2 Comparison Against the State of the Art

5.2.1 Data Sets

For our experiments, we use four different popular data sets often used in fair classification (Table 2):

- *Adult* The Adult UCI income data set [25] contains 14 demographic attributes of approximately 45,000 individuals together with class labels which state whether their income is higher than $50,000 or not. As sensitive attribute, we use gender encoded as a binary attribute, male or female.
- *COMPAS* The COMPAS data set [3] contains 13 attributes of about 7,000 convicted criminals with class labels

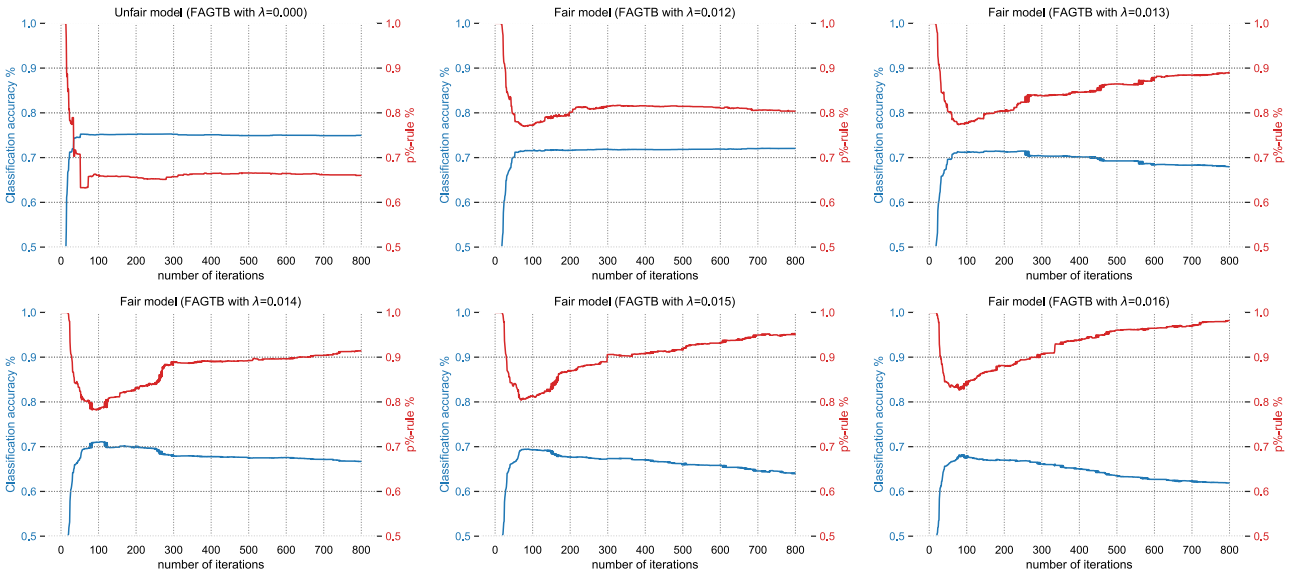

Fig. 2 Synthetic scenario: accuracy and p-rule metric for a biased model ($\lambda = 0$) and for several fair models with varying values of λ optimized for demographic parity

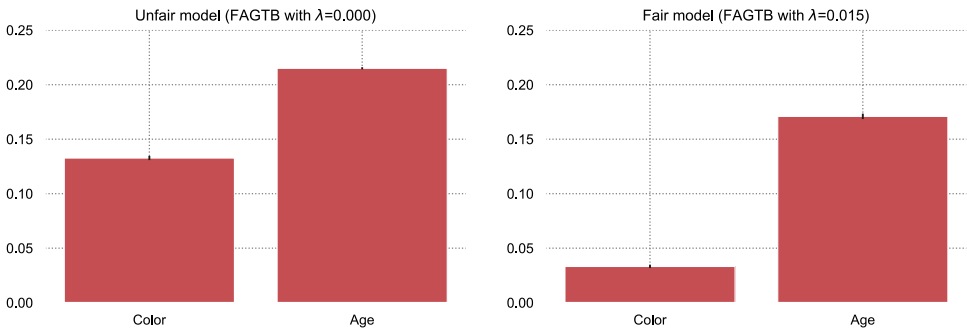

Fig. 3 Synthetic scenario: feature importance for a biased model ($\lambda = 0$) and a fair model ($\lambda = 0.015$) optimized for demographic parity

Table 2 Data sets used for the experiments

Data set	# Observations	# Features	Target	%Target	Sensitive	%Sensitive
Adult UCI	45,000	14	Income >= $50k	30.0	Gender	58.0
COMPAS	6967	13	2-year recidivism	45.5	Race	34.0
Default	30,000	23	Defaulting on payments	22.1	Gender	60.4
Bank	45,211	16	Subscription to a term deposit	11.7	Age	32.9

Description of the data sets: number of observations, number of features, target, total share of the target, sensitive attribute and total share of the sensitive attribute

that state whether or not the individual recidivated within 2 years of its most recent crime. Here, we use race as sensitive attribute, encoded as a binary attribute, Caucasian or not Caucasian.

- *Default* The Default data set [26] contains 23 features about 30,000 Taiwanese credit card users with class labels which state whether an individual will default on payments. As sensitive attribute we use gender encoded as a binary attribute, male or female.
- *Bank* The bank marketing data set [27] contains 16 features about 45,211 clients of a Portuguese banking institution. The goal is to predict whether the client has subscribed or not to a term deposit. We consider the age as sensitive attribute, encoded as a binary attribute, indicating whether the client's age is between 33 and 60 years or not.

For all data sets, we repeat ten 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.

5.2.2 Fairness Algorithms

Because different optimization objectives result in different algorithms, we run separate experiments for the two fairness metrics of our interest: demographic parity (Table 3) and equalized odds (Table 4). 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 whether 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.

In addition to the algorithms mentioned above, we evaluate the following fair state-of-the-art in-processing algorithms: Wadsworth [18][2], Zhang [17][3], Kamishima [28][1] Feldman [8][1], Zafar-DI [29][1] and Zafar-DM [10][1].
[123]

For each algorithm and for each data set, we obtain the best hyperparameters by grid search in fivefold cross-validation (specific to each of them). As a reminder, for FAGTB the λ value is used to balance the two 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 Fig. 4 for the Adult UCI data set. For that, we model the FAGTB-NN algorithm with ten different values of λ and we run each experiment ten 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 optimization with a binary cross-entropy loss. For the Adult UCI data set for example, the architecture consists of two hidden layers with 16 and eight units, respectively, and ReLU activations. The output layer comprises one single output node with sigmoid activation.

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

[1] https://github.com/algofairness/fairness-comparison.

[2] https://github.com/equialgo/fairness-in-ml.

[3] https://github.com/IBM/AIF360.

Table 3 Results for demographic parity

	Adult		COMPAS		Default		Bank	
	Accuracy (%)	*P*-rule (%)	Accuracy (%)	*P*-rule (%)	Accuracy (%)	*P*-rule (%)	Accuracy (%)	*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
Wadsworth 2018 [18]	83.1	89.7	63.9	90.1	81.8	90.0	**90.2**	90.1
Zhang 2018 [17]	83.3	90.0	64.1	89.8	81.4	90.0	90.0	90.0
Zafar-DI [14]	82.2	89.8	63.9	89.7	80.7	89.8	89.2	90.1
Kamishima [28]	82.3	89.9	63.8	90.0	81.1	90.0	89.6	89.9
Feldman [8]	–	–	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

The results in bold represent the best performance achieved for each columns among the fair algorithms

Table 4 Results for equalized odds

	Adult			COMPAS		
	Accuracy (%)	DispFPR	DispFNR	Accuracy (%)	DispFPR	DispFNR
Standard GTB	86.8	0.06	0.07	69.1	0.12	0.20
Standard NN	85.3	0.07	0.10	67.5	0.09	0.15
FAGTB-1-Unit	86.3	0.02	0.02	65.1	0.03	0.12
FAGTB-NN	**86.4**	0.02	0.02	**66.2**	0.01	0.02
Wadsworth 2018 [18]	84.9	0.02	0.03	65.4	0.02	0.03
Zhang 2018 [17]	84.8	0.03	0.03	64.9	0.03	0.02
Zafar-DM [10]	83.9	0.03	0.09	64.3	0.09	0.17
Kamishima [28]	82.6	0.06	0.24	63.6	0.08	0.11
Feldman [8]	80.6	0.07	0.05	61.1	0.03	0.03
	Default			Bank		
	Accuracy (%)	DispFPR	DispFNR	Accuracy (%)	DispFPR	DispFNR
Standard GTB	82.9	0.02	0.04	90.8	0.04	0.06
Standard NN	82.1	0.02	0.05	90.3	0.05	0.08
FAGTB-1-Unit	82.1	0.00	0.01	89.7	0.02	0.07
FAGTB-NN	**82.5**	0.00	0.01	**90.3**	0.01	0.07
Wadsworth 2018 [18]	81.2	0.01	0.02	89.4	0.01	0.07
Zhang 2018 [17]	81.9	0.00	0.01	89.8	0.00	0.07
Zafar-DM [10]	81.0	0.00	0.03	89.5	0.01	0.08
Kamishima [28]	80.5	0.00	0.04	89.3	0.00	0.08
Feldman [8]	71.8	0.02	0.02	87.1	0.05	0.06

Comparing our approach with different common fair algorithms by accuracy and fairness (DispFPR, DispFNR) for the Adult UCI, the COMPAS, the Default and the Bank data set

The results in bold represent the best performance achieved for each columns among the fair algorithms

quickly, this may result in biased prediction probabilities during the first iterations which are difficult to correct by the adversary afterward. For FAGTB-NN, in order to achieve better results, we execute for each gradient boosting iteration 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

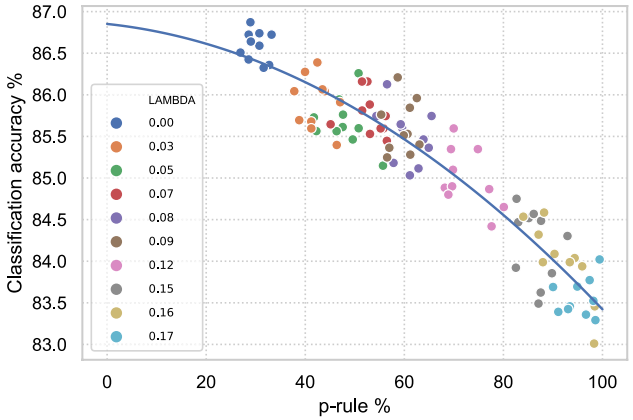

Fig. 4 Impact of hyperparameter λ (Adult UCI data set): higher values of λ produce fairer predictions, while λ near 0 allows to only focus on optimizing the classifier predictor

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.

5.2.3 Results

For demographic parity (Table 3), 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 Fig. 5, we plot the distribution of the predicted probabilities for each sensitive attribute S for three different models: an unfair model with $\lambda = 0$ and two 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.

Zhang [17] 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 Wadsworth [18] algorithm follows the same approach but without projection term and obtains similar results.

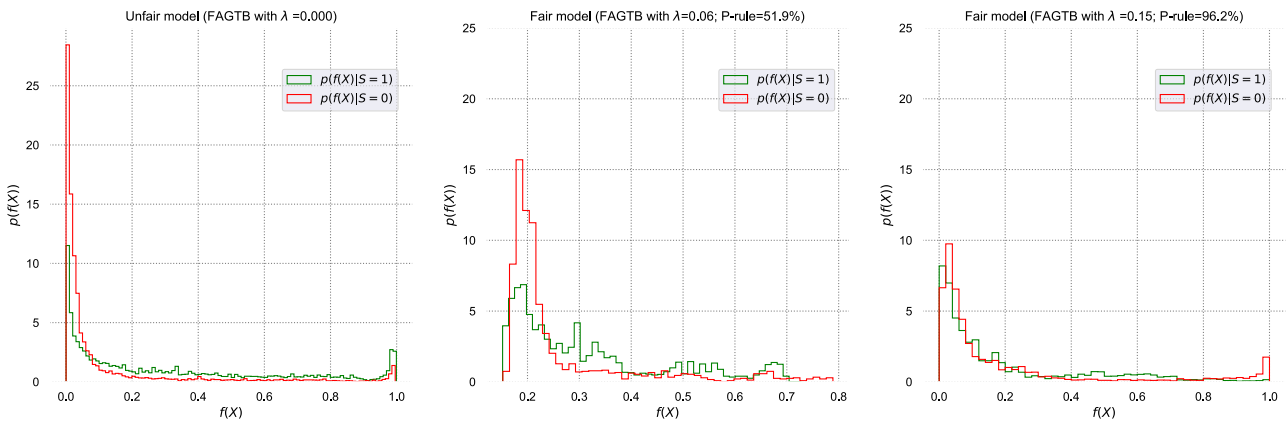

Fig. 5 Distributions of the predicted probabilities given the sensitive attribute S (Adult UCI data set)

For equalized odds, the min–max optimization is more difficult to achieve than demographic parity. The fairness metrics DispFPR and DispFNR 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 four data sets and for both metrics values below 0.02 or less, except for the Bank data set where DispFNR is equal to 0.07. For this data set, most of the state-of-the-art algorithms result in a DispFNR 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.

6 Conclusion

In this work, we developed a new approach to produce fair predictors, based on generic, non-necessarily differentiable, machines. Our gradient boosting framework indeed allows us to consider any regression machine, by iteratively feeding it with both prediction and fairness residuals as target outputs. This enables the use of very effective machines such as CART decision trees for fair machine learning. Compared with other state-of-the-art algorithms, our fair gradient tree boosting approach proves to be more efficient in terms of accuracy while obtaining a similar level of fairness. Currently, we use a neural network architecture for the adversary. We chose this approach in order to recover the gradient of the input. Another possible strategy is to replace the adversarial neural network by deep neural decision forests [30] which allow to recover the gradient by derivative. Such an architecture would therefore only be composed of trees. Another field left for further investigations is the mathematical identification of the optimal hyperparameter λ. Objectives here are a better convergence of the algorithm and the optimization of the trade-off between accuracy and fairness. Additionally, a recent work in [31] proposes a new hierarchical rule-based model for classification tasks, concept rule sets (CRS), with a strong transparent inner structure. With the aim of developing a model which achieves three objectives : a high classification performance, a low complexity and fair predictions, it would be interesting to implement this contribution with an adversarial neural network architecture. By taking up the general idea of our framework, the negative gradient from an adversarial which predicts the sensitive feature at each step could be added to the predictor gradient of the discrete CRS via continuous multilayer logical perceptron (MLLP) and random binarization (RB). Finally, it might be interesting to investigate a measure which does not only consider the general case of bias but can also spot and quantify bias that persists on specific subsegments of the population.

References

1. Zhang C, Liu C, Zhang X, Almpanidis G (2017) An up-to-date comparison of state-of-the-art classification algorithms. Expert Syst Appl 82:128–150
2. Bolukbasi T, Chang K-W, Zou JY, Saligrama V, Kalai AT (2016) Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In: Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R (eds) Advances in neural information processing systems, vol 29. Curran Associates Inc., Red Hook, pp 4349–4357
3. Angwin J, Larson J, Mattu S, Kirchner L (2016) Machine bias. ProPublica
4. Lambrecht A, Tucker CE (2018) Algorithmic bias? An empirical study into apparent gender-based discrimination in the display of STEM career ads. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2852260
5. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Official J Eur Union, vol L119, pp 1–88, May 2016
6. Pedreshi D, Ruggieri S, Turini F (2008) Discrimination-aware data mining. In: Proceeding of the 14th ACM SIGKDD international conference on knowledge discovery and data mining—KDD 08, p 560
7. Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R (2011) Fairness through awareness
8. Feldman M, Friedler S, Moeller J, Scheidegger C, Venkatasubramanian S (2014) Certifying and removing disparate impact, pp 1–28
9. Hardt M, Price E, Srebro N (2016) Equality of opportunity in supervised learning, pp 1–22
10. Zafar MB, Valera I, Rodriguez MG, Gummadi KP (2017) Fairness beyond disparate treatment and disparate impact, pp 1171–1180
11. Kamiran F, Calders T (2012) Data preprocessing techniques for classification without discrimination. In: Knowledge and information systems, vol 33, no 1, pp 1–33. https://doi.org/10.1007/s10115-011-0463-8
12. Bellamy RKE, Dey K, Hind M, Hoffman SC, Houde S, Kannan K, Lohia P, Martino J, Mehta S, Mojsilovic A, Nagar S, Ramamurthy

KN, Richards J, Saha D, Sattigeri P, Singh M, Varshney KR, Zhang Y (2018) AI fairness 360: an extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias
13. Calmon FP, Wei D, Ramamurthy KN, Varshney KR (2017) Optimized data pre-processing for discrimination prevention, pp 1–18
14. Zafar MB, Valera I, Rodriguez MG, Gummadi KP (2015) Fairness constraints: mechanisms for fair classification, vol 54
15. Celis LE, Huang L, Keswani V, Vishnoi NK (2018) Classification with fairness constraints: a meta-algorithm with provable guarantees. CoRR arXiv:1806.06055
16. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (eds) Advances in neural information processing systems, vol 27. Curran Associates, Inc., pp 2672–2680
17. Zhang BH, Lemoine B, Mitchell M (2018) Mitigating unwanted biases with adversarial learning. In: Association for the advancement of artificial intelligence
18. Wadsworth C, Vera F, Piech C (2018) Achieving fairness through adversarial learning: an application to recidivism prediction
19. Louppe G, Kagan M, Cranmer K (2016) Learning to pivot with adversarial networks. In: NIPS
20. Chen J, Kallus N, Mao X, Tech C, Svacha G (2019) Fairness under unawareness: assessing disparity when protected class is unobserved
21. Donini M, Ben-David S, Pontil M, Shawe-Taylor J (2017) An efficient method to impose fairness in linear models. In: NIPS workshop on prioritising online content
22. Friedman J (2001) Full-text. Ann Stat 29(5):1189–1232
23. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial networks, pp 1–9
24. Kusner MJ, Loftus JR, Russell C, Silva R (2017) Counterfactual fairness. In: NIPS
25. Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed 14 Aug 2019
26. Yeh I-C, Lien C-H (2009) The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Syst Appl 36(2):2473–2480
27. Moro S, Cortez P, Rita P (2014) A data-driven approach to predict the success of bank telemarketing. Decis Support Syst 62:06
28. Kamishima T, Akaho S, Asoh H, Sakuma J (2012) Fairness-aware classifier with prejudice remover regularizer. In: Flach PA, De Bie T, Cristianini N (eds) Machine learning and knowledge discovery in databases. Springer, Berlin, pp 35–50
29. Zafar MB, Valera I, Rodriguez M, Gummadi K, Weller A (2017) From parity to preference-based notions of fairness in classification, pp 229–239
30. Kontschieder P, Fiterau M, Criminisi A, Rota Bulo S (2015) Deep neural decision forests, pp 1467–1475
31. Wang Z, Zhang W, Liu N, Wang J (2020) Transparent classification with multilayer logical perceptrons and random binarization. In: Proceedings of the AAAI conference on artificial intelligence