On the Long-term Implications of Algorithmic Fairness Interventions: Mapping the Discourse to Affirmative Action Policies

Dan Kluser Dr. Hoda Heidari Prof. Andreas Krause

Abstract

Existing notions of fairness for Machine Learning ensure that algorithmic predictions satisfy some notion of error equality at the time of decision-making, but the long-term implications of these fairness constraints on society are not adequately studied. In this work, we analyze the long-term consequences of algorithmic interventions taking inspiration from the economic literature on *affirmative action policies*. We present a flexible *data-driven* framework that enables practitioners to better understand the potential societal consequences of a predictive model under a wide range of settings and circumstances. We illustrate our framework on a credit lending data set, and observe that not only different fairness constraints lead to very different societies in the long run, how we impose a given notion of fairness (e.g., through pre-, in-, or post-processing methods) can significantly alter the subjects' incentives to invest in qualifications. Our work is the first to systematically investigate and compare the long term implications of various fairness notions and fairness enhancing mechanisms, and it offers a practical toolkit for studying the human-level ramifications of decision-making models in a broad range of application domains.

1 Introduction

Fairnesss for Machine Learning has received considerable attention, recently. Predictive models—trained by learning algorithms on massive data sets of historical records—are increasingly employed to make highly consequential decisions for human subjects, in areas such as credit lending [Petrasic *et al.*, 2017], policing [Rudin, 2013], criminal justice [Barry-Jester *et al.*, 2015], employment [Miller, 2015], and medicine [Hart, 2017]. Decisions made in this fashion have long-lasting impact on people's lives and may adversely affect certain individuals or social groups [Sweeney, 2013; Angwin *et al.*, 2016]. This realization has recently spawned an active area of research into quantifying and guaranteeing fairness for machine learning [Dwork *et al.*, 2012; Kleinberg *et al.*, 2016; Hardt *et al.*, 2016].

Most existing notions of fairness guarantee some form of *allocative equality* at the time of decision making, but do not account for the *adverse impact* of algorithmic decisions today on the *long term* welfare and prosperity of different segments of the population. For instance, consider statistical parity. The notion requires that the model results in similar selection rate (i.e., equal percentage of positive predictions) across all social groups of interest. But it does not take into consideration the fact that in the long run, this equality constraint may incentivize different segments of the population to invest at different levels and in potentially different sets of qualifications—some of which might be socially and economically more desirable than others. This may in effect lead to further *marginalization* of these groups.

Motivated by these concerns about existing notions of fairness, we argue for a broader view of algorithmic models—one that treats them as *policies* with the potential of animating individuals and reshaping society over time. Among other considerations[1], such view of decision-making models necessitates a firm understanding of how individual decision subjects may *respond* to these models and how those responses may translate into *adverse impact* for certain segments of the population.

To formulate the long-term impact of algorithmic policies on the underlying population, we take inspiration from the economic literature on *affirmative action policies* [Coate and Loury, 1993b,a;

[1] Another important factor is how a utility maximizing decision maker—employing the model—would respond to its predictions. For instance, they may interpret the predictions in a certain way, or update the model entirely. Prior work [Liu *et al.*, 2018; Kannan *et al.*, 2019] has already addressed some of these considerations.

Chung, 2000]. Affirmative action quotas have long been proposed and implemented as *temporary* remedies to eliminate group-level inequalities in areas such as employment and education. Economic models have established at least two salient ways in which affirmative action policies can benefit the historically disadvantaged group. First, enforcing quotas can provide the qualified members of the disadvantaged group with the opportunity to demonstrate their abilities and counter existing negative stereotypes against the group. Second, a larger representation of the disadvantaged group in higher social positions can generate *role models* that positively influence future generations of the group in their investment decisions. At the same time, economic models also highlight the potential negative impact of affirmative actions on the group it aims to assist. Quota constraints effectively lower the bar for the disadvantaged group members, and this lower bar may in turn reduce their incentives to invest in costly qualifications. When unqualified members of the disadvantaged group are assigned to high-skilled tasks, they under-perform and their poor performance bolsters negative stereotypes against the group.

In this work, we combine and adapt existing economic models of statistical discrimination and affirmative action to study the long-term impact of fair machine learning. We replace the assumption of a Bayesian decision maker—basing its decisions on a single-dimensional signal and group membership—with a *learning* one—observing a multi-dimensional feature vector. We assume decision subjects respond to the model by making investment decisions so as to maximize their utility—where utility is defined as the additional benefit the individual subject can earn by updating their qualifications minus the cost of the modification.

We present a flexible data-driven framework that allows practitioners to investigate the potential impact of their predictive models on society under a wide variety of circumstances. Our simulations allow for a variety of choices for the cost function, benefit function, and number of rounds of interactions. We illustrate our framework in a simplified setting. Importantly, we observe that different fairness constraints may shift the group-conditional distribution of qualifications in vastly different directions. Not only various fairness constraints lead to very different societies in the long run, how we impose a particular notion of fairness (i.e., through pre-, in-, or post-processing methods) can significantly alter people's incentives to invest.

Our work is the first to systematically investigate and compare the long term impact of various fairness notions and fairness enhancing mechanisms. Our work raises a number of important questions about the formulation of fairness and the means through which they are enforced. Should we think of fairness interventions as perpetual or temporary remedies to eliminate existing inequalities once and for all? If the latter, which notions of fairness are most effective in the long run? How do algorithmic interventions compare with other types of interventions (e.g., subsidizing investment in qualifications before individuals are subject to algorithmic decision making)?

1.1 Related Work: Fairness for Machine Learning

To guarantee fairness for Machine Learning the first step has to be defining what (un)fairness precisely means. Many different notions have been proposed, and much of them are devoted to quantifying *statistical-* or *group*-level fairness. Group fairness notions require that given a classifier, a certain fairness metric is equal across all protected groups. Different choices for the metric have led to different naming of the corresponding fairness notions (see e.g., demographic parity [Kleinberg *et al.*, 2016; Dwork *et al.*, 2012; Corbett-Davies *et al.*, 2017], disparate impact [Zafar *et al.*, 2017; Feldman *et al.*, 2015], equality of odds [Hardt *et al.*, 2016], and calibration [Kleinberg *et al.*, 2016]).

Several different mechanisms have been proposed to guarantee fairness, ranging from pre-processing of the training data, to in-processing, which imposes fairness constraints during training, to post-processing of trained-model's predictions [Berk *et al.*, 2017]. Pre-processing methods are based on the premise that removing biases from the training data will automatically result in a fair predictive model [Kamiran and Calders, 2009; Calders *et al.*, 2009; Hajian *et al.*, 2011]. In-processing methods make certain assumptions about the distribution of predictions made by a fair model, then make sure the trained model reflects those ideal fairness desiderata [Agarwal *et al.*, 2018]. The hope with in-processing methods is that imposing a fairness constraint during training would guarantee that the model is fair towards future instances. Post-processing methods similarly require certain equalities in

the distribution of algorithmic predictions, but ensure those constraint by tweaking the predictions of a black-box, potentially unfair predictive model [Hardt *et al.*, 2016].

Several recent papers study the long-term impact of decision-making models and fairness interventions on society and individuals (see, e.g., [Liu *et al.*, 2018; Kannan *et al.*, 2019]). Unlike prior work, our focus is on *how subjects respond* to algorithmic policies by *improving/updating their qualifications*. We don't make any case-specific assumptions about how the world changes in response to the deployed model, rather allow our micro-scale behavioral model to derive the macro-level change.

Also related but orthogonal to our work is a recent line of research on *strategic classification*—a setting in which decision subjects are assumed to respond *strategically* and potentially *untruthfully* to the choice of the classification model, and the goal is to design classifiers that are robust to strategic manipulation [Dong *et al.*, 2018; Hu *et al.*, 2019; Milli *et al.*, 2019].

Similar to our work is [Hu and Chen, 2018] take inspiration from existing models of statistical discrimination and affirmative action, and study the impact of enforcing statistical parity on hiring decisions made in a temporary labor market that precedes the permanent labor market. They show that under certain conditions, statistical parity can result in an equilibrium that Pareto-dominates the one that would emerge in an unconstrained labor market.

Two recent papers focus the on dynamics through which decision making models impact individuals. [Heidari *et al.*, 2019] study the long-term impact of decision making models by simulating social learning dynamics among decision subjects. They assume that individuals respond to the decision making model by imitating their role model—someone similar to them who has received higher benefit.

[Mouzannar *et al.*, 2019] reduce a predictive model to its selection rates across qualified an unqualified members of two socially salient groups (that is, 4 positive numbers). They assume that there exist continuously differentiable functions f_1 and f_2 that map these selection rates to the percentage of positive individuals in each group. Unlike their work, we don't focus solely on how true labels change in response to the decision making model; we also look at how the feature distribution change in response.

2 Translation to Fair ML

In this Section, we propose a variant of affirmative action models tailored to the study of automated decision making. Consider an environment with a decision maker and individual subjects sampled from an unknown distribution. The game consists of multiple stages. At a high-level, in the initial stage, the decision maker trains a decision making model using a training data set sampled from the initial population. This decision making model is then deployed to make decisions for individual subjects. In the next stage, decision subjects respond to the model by updating their qualifications. This changes the underlying population to what we refer to as the impacted population. A decision subject's goal is to make the investment decision that maximizes his/her utility. The decision maker's goal is to trains a model that maximizes its accuracy on the impacted population.

We model the initial population as distribution $\mathcal{D}_0$ defined over $\mathcal{S} \times \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the space of all possible feature vectors/qualifications, $\mathcal{Y}$ is the space of all possible true labels, and $\mathcal{S}$ specifies socially salient groups of interest. For simplicity, we assume individuals belong to one of two identifiable groups (i.e., $\mathcal{S} = \{B, W\}$).

In the first stage, the decision maker employs a (supervised) learning algorithm to train a decision making model. The learning algorithm receives a training data set $D_0 = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ consisting of n instances, where $\mathbf{x}_i \in \mathbb{R}^k$ specifies the feature vector for individual i and $y_i \in \mathcal{Y}$, his/her true label. Let $s_i \in \mathcal{S}$ specify the sensitive feature/ group membership for individual i. For example, s_i could specify race. For simplicity and unless otherwise specified, we assume the sensitive feature is excluded from $\mathbf{x}_i$. Instances in D_0 are sampled i.i.d. from the initial population $\mathcal{D}_0$.

The learning algorithm uses the training data D_0 to fit a decision-making *model* (or a hypothesis) $h : \mathcal{X} \to \mathcal{Y}$. Let $\mathcal{H}$ be the hypothesis class consisting of all the models the learning algorithm can choose from. A learning algorithm receives the training data D_0 as the input; then utilizes the data to select a model $h \in \mathcal{H}$ that minimizes some notion of empirical loss, $\mathcal{L}(D_0, h)$. For instance, in

classification the cost-sensitive 0-1 loss of a model h on the training data D_0 is defined as

$$\mathcal{L}(D_0, h) = \sum_{i=1}^{n} \mathcal{L}(h(\mathbf{x}_i), y_i) = \sum_{i=1}^{n} c_{FP}\mathbf{1}[y_i < h(\mathbf{x}_i)] + c_{FN}\mathbf{1}[y_i > h(\mathbf{x}_i)].$$

When the trained model h in reference is clear from the context, we denote individual i's predicted label by $\hat{y}_i$ (i.e., $\hat{y}_i = h(\mathbf{x}_i)$).

2.1 Benefits and Costs

We assume there exists a benefit function $b : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \mathbb{R}$, such that $b(\mathbf{x}, y, h)$ specifies the benefit/harm an individual with feature vector $\mathbf{x}$ and true label y receives as the result of being subject to the decision making model h. For simplicity and unless otherwise specified, we assume $b(\mathbf{x}, y, h) = h(\mathbf{x})$.

Individuals have investment cost c, which is distributed in the population according to the cumulative distribution function $G_s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$. Note that the cost distribution can depend on group identity s. Individuals must decide—in response to the decision making model—whether investing in qualifications is worthwhile. This depends on the extent to which investing raises the chance of being predicted as positive. The rational individual invests if the cost of investment does not exceed its expected benefit. More precisely, he/she would update his qualifications from $\mathbf{x}$ to $\mathbf{x}'$ to maximize his/her utility defined as additional benefit minus cost: $b(\mathbf{x}', h) - b(\mathbf{x}, h) - c(\mathbf{x}, \mathbf{x}')$.

Nearest neighbor matching to extrapolate the true labels.

In summary, the timing of the game is as follows. In Stage 0, Nature draws n individuals from D_0, along with their investment costs c from the group-dependent distribution G_s. In Stage 1, the decision maker trains a model h using D_0. Finally, in stage 2, individuals will respond to the model h by updating their qualifications.

3 Simulation Dynamics

3.1 Agent's Best Response

We begin by introducing a crucial definition. Recall that the agents update their feature vectors (qualifications) in response to the model h in Stage 2. The model h is the decision makers best-response from Stage 1.

Definition 1 (Agent's best-response) *For some agent i with d-dimensional feature vector $\mathbf{x}$ and group membership s,*

$$\arg\max_{\mathbf{x}' \in D^i} U_{\mathbf{x},h}(\mathbf{x}') = \arg\max_{\mathbf{x}' \in D^i} b(\mathbf{x}', h) - b(\mathbf{x}, h) - c_s(\mathbf{x}, \mathbf{x}')$$

is the best-response to the hypothesis h given cost function c_s and benefit function b.

Where $D^i = D^i_1 \times D^i_2 \times ... \times D^i_d$ and D^i_j is the domain of feature j for agent i. If feature j is immutable, then $D^i_j = \{x_i\}$. Note that the domain may be further constrained by the cost function.

We approximate the optimal solution using a gradient based optimization. To do so we need to further constrain the above optimization goal by requiring the benefit function b and cost function c_s to be differentiable.

3.1.1 Gradient Ascend

The optimization problem we arrive at is non-linear, non-convex and contains a mixture of integer and real variables. We begin our description of the gradient ascend algorithm by introducing some notation.

Notation		
$\mathbf{X}$	$\mathbb{R}^{n \times d}$	Feature matrix containing d features for each of the n agents.
$\mathbf{X}'$	$\mathbb{R}^{n \times d}$	Current estimation of best-response
$U_{\mathbf{X},h}$	$\mathbb{R}^{n \times d} \to \mathbb{R}^n$	Utility function
$\nabla U_{\mathbf{X},h}$	$\mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}$	Utility function's gradient evaluated for each row of its argument
∇	$\mathbb{R}^{n \times d}$	Matrix containing current gradient approximation
ϵ	$\mathbb{R}$	Used as threshold for stopping criteria. Magnitude of all components of gradient close to zero ($< \epsilon$) indicates (local) optima.
α	$\mathbb{R}$	Learning rate
maxit	$\mathbb{N}$	Maximum number of iterations for the gradient ascend
it	$\mathbb{N}$	Counter for number of iterations
scale_dummycoded	$\mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}$	Scales dummy-coded features such that the dummy-coded representation always sums to one. (Section 3.1.3)
clip	$\mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}$	Clips feature values outside their domain to ensure feasibility of changes (Section 3.1.4)
discretize	$\mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}$	Maps the continuous values assigned to discrete variables to the closest discrete value. (Section 3.1.4)

Algorithm 1: Gradient Ascend

Data: $\mathbf{X} \in \mathbb{R}^{n \times d}$, $U_{\mathbf{X},h}(\mathbf{X}') = b(\mathbf{x}', h) - b(\mathbf{x}, h) - c_s(\mathbf{X}, \mathbf{X}')$
Result: $\mathbf{X}^*$ containing best-responses for all agents

1 $\mathbf{X}' \leftarrow \mathbf{X}$;
2 $\nabla \leftarrow \nabla U_{\mathbf{X},h}(\mathbf{X}')$;
3 **while** $it < maxit$ *and* $\exists i, j \, |\nabla_{ij}| > \epsilon$ **do**
4 $\quad \mathbf{X}' \leftarrow \mathbf{X}' + \alpha \cdot \nabla$;
5 $\quad \mathbf{X}' \leftarrow$scale_dummycoded($\mathbf{X}'$);
6 $\quad \mathbf{X}' \leftarrow$clip($\mathbf{X}'$);
7 $\quad \nabla \leftarrow \nabla U_{\mathbf{X},h}(\mathbf{X}')$;
8 $\quad it \leftarrow it + 1$;
9 $\mathbf{X}' \leftarrow$ discretize($\mathbf{X}'$) ;
10 **for** $i = 1$ **to** n **do**
11 $\quad$ **if** $U_{\mathbf{X},h}(\mathbf{X}')_i > 0$ **then** $\mathbf{X}^*{}_i \leftarrow \mathbf{X}'{}_i$;
12 $\quad$ **else** $\mathbf{X}^*{}_i \leftarrow \mathbf{X}_i$;

Description In lines 1 and 2 initialization is performed. We initialize ∇ by setting it to the gradient of $U_{\mathbf{X},h}$ at $\mathbf{X}'$ in line 2. The main loop on line 3 either stops after *maxit* iterations or if every component of the gradient is sufficiently small ($< \epsilon$, for some small ϵ). We then update the current estimate $\mathbf{X}'$ by $\alpha \cdot \nabla$, where α is our learning rate and ∇ points towards the direction of the steepest ascend. On line 7 the gradient is calculated for the new estimate $\mathbf{X}'$. After termination of the main loop, we calculate for every agent the utility $U_{\mathbf{X},h}(\mathbf{X}')_i$ ($\mathbf{X}'{}_i$ is the approximated best-response for agent i). If this best-response has a utility strictly larger than zero, the agent sets its own feature vector to $\mathbf{X}'{}_i$ otherwise the agent's feature vector remains $\mathbf{X}_i$. A utility smaller than zero may occur due to overshooting, which is a common phenomena with gradient based optimization techniques.

Lines 5, 6 and 9 are concerned with enabling gradient ascend on dummy-coded (one hot encoded) features, keeping features inside their bounds as defined in their domains D_j^i and handling discrete features. Refer to the following sections for an in-depth discussion.

3.1.2 Gradient Estimation

We aim to support as many combinations of benefit and cost functions as possible within the framework. Thus, instead of analytically deriving the gradient, a finite difference approximation is used.

Definition 2 (Central Difference Approximation)

$$(\nabla_t f)(\mathbf{x})_j = \frac{f(\mathbf{x} + \mathbf{e}_j \cdot \frac{t}{2}) - f(\mathbf{x} - \mathbf{e}_j \cdot \frac{t}{2})}{t}$$

Where $f : \mathbb{R}^d \to \mathbb{R}$ is some differentiable function, $\mathbf{e}_j$ refers to the unit vector with all components set to zero except for the j'th component, $\mathbf{x} \in \mathbb{R}^d$, $j \in \mathbb{N}, j \leq d$ and $t \in \mathbb{R}$.

Using the Taylor expansion of some differentiable function f, it is easily verified that

$$\lim_{t \to 0} |(\nabla_t f)(\mathbf{x}) - (\nabla f)(\mathbf{x})| = \lim_{t \to 0} O(d \cdot t) = 0$$

Thus, for a very small t we can expect a similarly small error in our approximation.

3.1.3 One-hot-encoded Features

One-hot encoding is often used for categorical features for which no meaningful partial order can be derived.

Example 1 (One-hot-encoding) *Let us assume that feature $\mathbf{x}_i$ has domain $D_i = \{A, B, C, D\}$. Further assume there is no natural partial order for the set D_i. One-hot-encoding maps this single feature to $|D_i|$ many binary features:*

$$\mathbf{x}_{i,A} = \mathbb{1}[\mathbf{x}_i = A] \quad \mathbf{x}_{i,B} = \mathbb{1}[x_i = B] \quad \mathbf{x}_{i,C} = \mathbb{1}[\mathbf{x}_i = C] \quad \mathbf{x}_{i,D} = \mathbb{1}[\mathbf{x}_i = D]$$

For ease of notation we will define one-hot(j) as the set of feature indices of the one-hot encoding of feature j.

One-hot-encoding increases the number of features while the number of permissible values remains the same. Thus one-hot-encoding creates coupled features, which poses a challenge for the gradient ascend in Algorithm 1. As the number of features increases, exhaustively testing all the permutations quickly becomes infeasible due to computational limitations.

Ideally we would like to have exactly one feature of the dummy-coded representation set to one and the others to zero upon termination of the gradient ascend. It is likely to happen that multiple features of the dummy-coded representation have a positive gradient in most places. If one of those features is initially set to one, the gradient ascend will never decrease it, even if the optimal solution would require this particular dummy-coded feature to be zero. Below algorithm, which is called on line 5 in Algorithm 1, mitigates this problem by scaling the dummy-coded representation such that the dummy-coded representation of one feature always sums to one.

Algorithm 2: Scale Dummycoded

Data: $\mathbf{X} \in \mathbb{R}^{n \times d}$
foreach $\mathbf{x} \in \mathbb{R}^d$ *in* $\mathbf{X}$ **do**
  foreach *dummy-coded feature* j **do**
    sum $\leftarrow \sum_{k \in \text{one-hot}(j)} \mathbf{x}_k$;
    foreach $k \in$ *one-hot*(j) **do**
      $\mathbf{x}_k \leftarrow \frac{\mathbf{x}_k}{\text{sum}}$;

After termination of the main loop of the gradient ascend, we might still have multiple non-zero features of the dummy-coded representation for one categorical feature. We address this by finding the maximum value for each dummy-coded representation and setting this to one and all the others to zero.

$$\mathbf{x}_{ik} = \begin{cases} 1 & \text{if } \exists j,\ k = \arg \max_{l \in \text{one-hot}(j)} \mathbf{x}_{il} \\ 0 & \text{otherwise} \end{cases}$$

We chose not to add a regularizer to enforce the above constraints relevant to the one-hot encoded features (their sum should equal 1 and their values should be 0 or 1). While this regularizer could be formalized easily, we would create a local optima at the initial dummy-coded feature values, which becomes a problem in conjunction with our local gradient ascend optimization approach.

3.1.4 Discrete Features

During the gradient ascend all features are assumed to be continuous. Upon termination of the gradient ascend there will be instances with features outside of their domain. We address this in two ways.

First, during the gradient ascend features are clipped (line 6 in Algorithm 1). Clipping in this context refers to ensuring that values stay within the bounds of their domains. This is done after every step of the gradient ascend because out of bound feature values influence the gradient of other features. Clipping therefore improves our approximation of the optimal solution. Denote the bounds for feature j and agent i as $a_j^i = \max_{d \in D_j^i} d$ and $b_j^i = \min_{d \in D_j^i} d$.

$$\mathbf{X}_{ij} = \min(a_j^i, \mathbf{X}_{ij})$$

$$\mathbf{X}_{ij} = \max(b_j^i, \mathbf{X}_{ij})$$

The function clip as called on line 6 of Algorithm 1, executes the above assignments for all agents i and features j.

Secondly, after the gradient ascend, discrete features might still not be valid, meaning $\mathbf{X}_{ij} \notin D_j^i$ for some agent i and feature j. Thus we simply assign the closest valid value to this feature.

$$\mathbf{X}_{ij} = \arg\min_{z \in D_j^i} |\mathbf{X}_{ij} - z|$$

The above assignment, done for all agents i and discrete features j, is done in the function discretize as called on line 9 in Algorithm 1.

3.2 Integration of Fairness Libraries

The gradient ascend approach used in our framework requires the utility function to be differentiable. Many existing tools to enforce fairness constraints do not meet this requirement out of the box. Luckily, many tools can be easily modified to output the necessary continuous scores.

3.2.1 Reweighing

Reweighing [Kamiran and Calders, 2012] as implemented in AIF360 is a pre-processing technique which enforces statistical parity by assigning weights to instances which are then used during training for some fairness unaware classifier.

Reweighing in combination with logistic regression is employed in Section 4 as a pre-processing technique for the evaluation of statistical parity.

3.2.2 Fair Learn

Fair Learn "reduces fair classification to a sequence of cost-sensitive classification problems". [Agarwal *et al.*, 2018] The implementation as provided by the authors yields a so called "best-classifier" (provably best in their setting). This "best-classifier" is a weighted combination of several classifiers:

$$h(\mathbf{x}) = \sum_{i=1}^{n} w_i \cdot \mathbb{1}[h_i(\mathbf{x}) > 0.5]$$

Clearly the classifier h is not differentiable. To see this, note that h may output at most 2^n distinct scores. Instead we use the obvious differentiable counterpart to the above "best-classifier":

$$h(\mathbf{x}) = \sum_{i=1}^{n} w_i \cdot h_i(\mathbf{x})$$

This amounts to a small non intrusive change in the Fair Learn library. The training procedure is left untouched. We use Fair Learn in Section 4 to enforce statistical parity as our in-processing technique in the evaluation of statistical parity.

3.2.3 Reject Option Classification

Reject option classification (ROC) [Kamiran *et al.*, 2012a] is a post-processing technique for binary classification to enforce several fairness constraints. We used the implementation from IBMs AIF360 framework [Bellamy *et al.*, 2018]. In order to satisfy some fairness constraint like statistical parity, reject option classification modifies predictions made by some fairness unaware classifier.

The intention is to keep the accuracy as high as possible, thus the algorithm relabels instances for which the uncertainty is highest or in other words, those with a score as close to the threshold as possible.

ROC finds some threshold $t \in [0, 1]$ and a margin $m \in [0, 1]$. Instances belonging to the unprivileged class having a score from the fairness unaware classifier within this critical region $[t - m, t + m]$ get relabeled from 0 to 1, where 1 is the favorable label. ROC also relabels privileged instances within the critical region $[t - m, t + m]$ from 1 to 0.

To use ROC's prediction as our benefit function b, we need a differentiable surrogate function approximating ROC's relabeling. We propose the following function, which maps scores to scores instead of scores to labels:

$$f(x, s) = x + (-1)^s \cdot (\sigma_\alpha(x - t^-) - \sigma_\alpha(x - t^+)) \cdot m$$

where $\sigma_\alpha(x) = \sigma(\alpha \cdot x)$ is the sigmoid function with its argument multiplied by some constant α, $t^- = t - m$, $t^+ = t + m$, x is the score from the fairness unaware classifier and s the protected group membership of the instance at hand. Intuitively, the difference of the sigmoid functions evaluates to 1 if and only if $x \in [t^-, t^+]$. Note that for the predicted label $\hat{y}$ we use ROCs prediction function.

For the experiments discussed in Section 4 ROC was used in the comparison of different notions of fairness to enforce equality of average odds and statistical parity. Also ROC is employed in Section 4 as a post-processing technique to enforce statistical parity in the comparison of different "times of intervention" (namely pre-, in- and post-processing).

3.2.4 Calibration

Definition 3 *We call a classifier h calibrated for groups iff.*

$$\forall s \in \{0, 1\}, \forall p \in [0, 1] \qquad \Pr_{(\mathbf{x},y)\sim\mathcal{D}}[y = 1 \mid h(\mathbf{x}) = p, \mathbf{x}_s = s] = p$$

where $x_s \in \{0, 1\}$ is the protected group membership. Our definition is similar to the one found in [Pleiss et al.*, 2017].*

For our evaluation of different notions of fairness we used a classifier satisfying the above definition. First, we fit a logistic regression on the whole dataset. Afterwards, calibration using platt scaling is done for each group individually, resulting in one classifier per group. Even though logistic regression is quite well calibrated by default, we think that calibrating it individually for each group will highlight the fairness aspect of calibration in our evaluation.

4 Experiments

In this Section, we illustrate our framework in a specific setting in which the benefit an individual receives is simply their prediction $b(\mathbf{x}, y, h) = h(\mathbf{x})$, and the cost function $c_s(\mathbf{x}, \mathbf{x}')$ is defined as proposed by Heidari *et al.* [2019]. We train a logistic regression model and impose different fairness constraints on it using various notions of group fairness and various mechanisms to enforce them. We ran our simulations on the German Credit data set using age as our protected group attribute. We used the implementations provided by the AIF 360 package [Bellamy *et al.*, 2018].

We conduct a series of experiments to answer the following questions:

- Does it matter which notion of fairness is imposed on the decision maker?
- For a fixed notion of fairness (namely, statistical parity) does it matter how we impose it (e.g., through pre-, in-, or post-processing methods?)

4.1 Setup

Recall that in *Stage 1* the decision maker comes up with her best response to the dataset. This best response is the model h minimizing the classifier's loss function on the dataset. This initial dataset, together with the scores given by the best-response classifier, is what we will call *initial*.

In *Stage 2* agents find their best-response using the extensively described gradient ascend algorithm. We refer to the resulting modified dataset, which contains the best responses of all agents and the corresponding scores (benefit) from h, as *impacted*.

When reasoning about the long-term impact of a particular fairness measure, we will consider two metrics: First, the difference between the median *privileged* (P) and median *unprivileged* (UP) benefits. Clearly it is desirable that the disparity between *UP* and *P* decreases from the *initial* to the *impacted* population.

We will also look at the distribution of specific features. Some fairness measures will incentivize certain groups to manipulate more (in a desirable direction) and some will disincentivize agents to manipulate. A fairness constraint should not leave some group worse off feature-wise (because they were disincentivize to manipulate) compared to the situation without any fairness constraint (referred to as *no constraint* in the following sections).

4.2 The Long-term Implications of Various Notions of Fairness

For comparability, all of the below methods are post-processing techniques applied to a fairness unaware logistic regression model. Meaning, we do not exclude the protected group attribute during training.

Statistical Parity We used the Reject Option post-processing technique as provided in AIF360, with minor modifications as described in Section 3.2.

Calibration Platt scaling is applied individually for both groups.

Equality of Average Odds We used the Reject Option post-processing technique as provided in AIF360, with minor modifications as described in Section 3.2.

From the benefit distribution comparison in Figure 1, we may conclude that *no constraint* seems to reduce disparity the most.

Although *calibration* initially increases disparity on our data set (compared to *no constraint*, the *disparity before* increases by 0.016, see Table 1), the medians of the distribution in the impacted distributions are closer together than before. This indicates that calibration, while providing fairness guarantees, might not disincentivize the unprivileged group to improve their features.

The picture painted by *statistical parity* and *average odds* is quite different. While for both of them the disparity decreases in the *initial* distribution, the disparity is much worse for the *impacted* distribution. Looking at the distribution of features *savings* and *purpose* in Figure 3, one can also see *that ROC seems to incentivize the unprivileged to manipulate less*. In more concrete terms, those fairness interventions leave the unprivileged instances worse-off in terms of their feature values.

This is mostly due to the post-processing technique used by ROC to enforce both *statistical parity* and equality of *average odds*. Recall that ROC gives a score boost to the unprivileged in a certain region. The upper bound of this region is a local optima. Due to the nature of our gradient ascend, instances belonging to the unprivileged group will not move beyond this local optima. Although mostly a property of our local optimization, we still believe this result might be of relevance as agents

in the real world are not likely to employ some kind of global optimization. Our result is similar in spirit to the patronizing equilibrium in [Coate and Loury, 1993b].

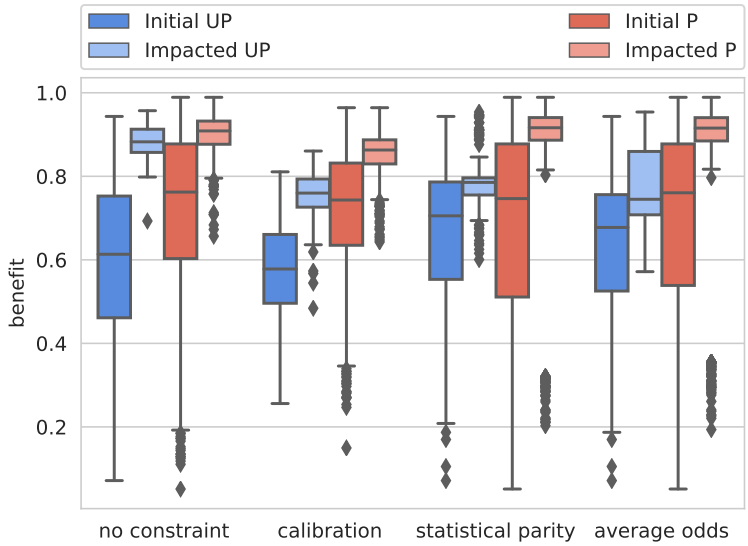


Figure 1: While the unconstrained model and calibration seem to reduce disparity, both fairness constraints enforced by ROC (*statistical parity* and *average odds*) seem to increase the disparity in our model.

	no constraint	calibration	average odds	statistical parity
Disparity before	0.149	0.165	0.083	0.041
Disparity after	0.026	0.103	0.170	0.131
Increase UP	0.269	0.182	0.067	0.08
Increase P	0.146	0.120	0.155	0.17

Table 1: Summary of the underlying data of Figure 1. For every notion of fairness (columns), the difference of medians between privileged and unprivileged is given both before and after the simulation. Refer to the bottom two rows to see by how much the median increased for each group.

4.3 The Long-term Implications of Pre-, In-, and Post-processing Mechanisms

For comparability, all of the below methods are applied in conjunction with a fairness unaware logistic regression model. Meaning, we do not exclude the protected group attribute during training.

Pre-processing Reweighing is a pre-processing technique that weighs the examples in each (group, label) combination differently to ensure statistical parity of true labels before training a classifier [Kamiran and Calders, 2012].

In-processing "Fair Learn" is an in-processing technique that simultaneously optimizes accuracy and some fairness constraint (in our case statistical parity). [Agarwal *et al.*, 2018]

Post-processing Reject option classification is a post-processing technique that gives favorable outcomes to unprivileged groups and unfavorable outcomes to privileged groups in a confidence band around the decision boundary with the highest uncertainty [Kamiran *et al.*, 2012b]

While the disparity in the benefit distribution (see Figure 2) decreases (compare *initial* and *impacted*) for *no constraint* and *pre*, for *in* there is a negligibly small increase, the disparity increases for *post*. Refer to Section 4.2 for a discussion of why the ROC post-processing leaves unprivileged agents

worse off. The feature distributions in Figure 4 confirm this trend. Judging from our experiments *statistical parity might best be enforced using pre- or in-processing techniques*. Further experiments into the stability of those results, specifically whether the observed situations are equilibria or not, are required.

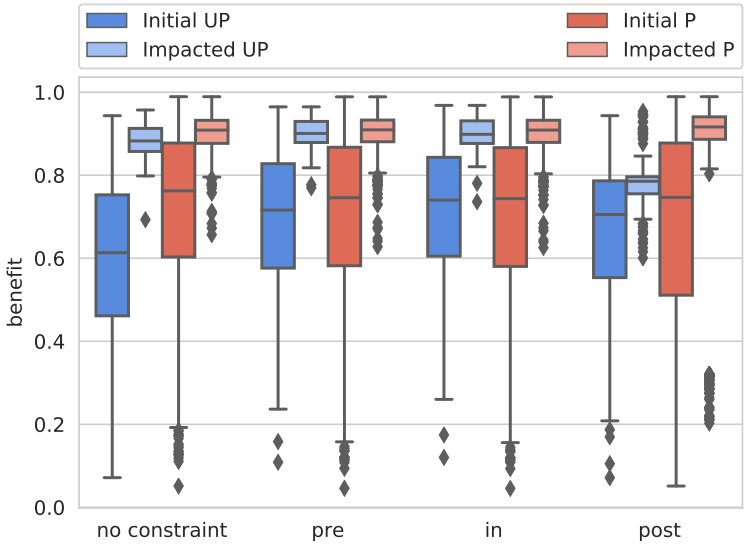
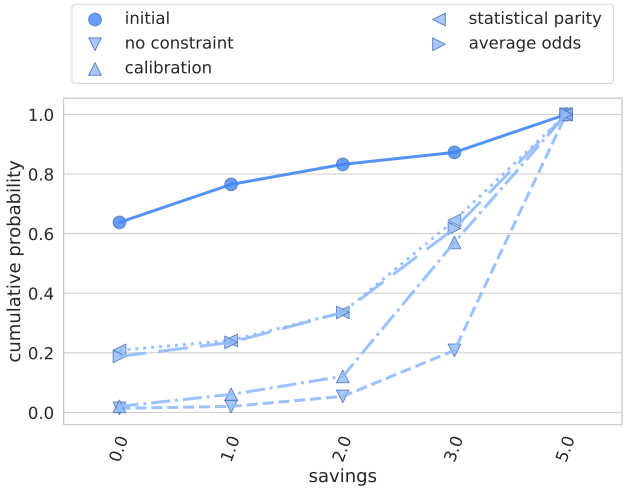


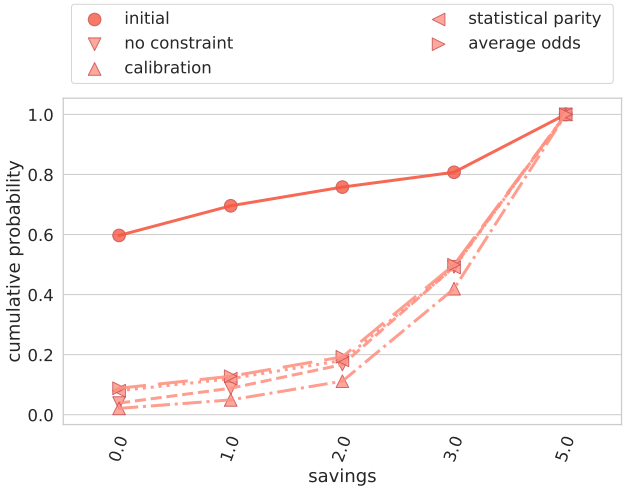
Figure 2: The unconstrained model, as well as pre- and in-processing, all decrease disparity in the impacted distribution. Post-processing increases disparity.

	no constraint	pre	in	post
Disparity before	0.149	0.030	0.004	0.041
Disparity after	0.026	0.009	0.010	0.131
Increase UP	0.269	0.185	0.158	0.08
Increase P	0.146	0.164	0.165	0.17

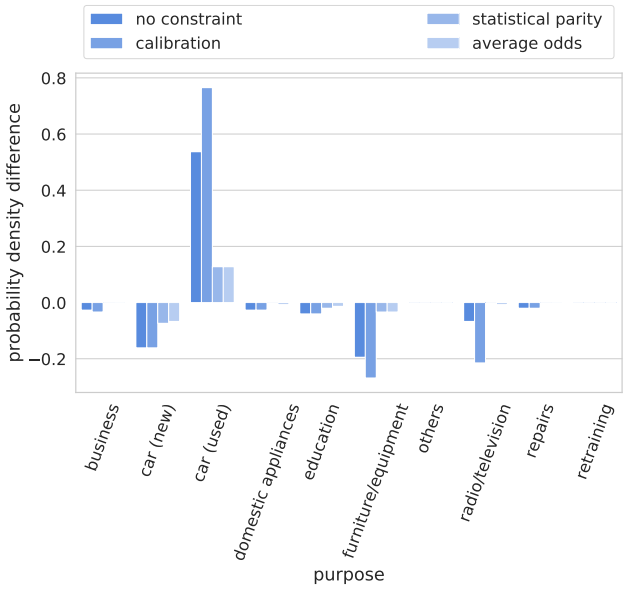
Table 2: Summary of the underlying data of Figure 2. For methods enforcing statistical parity (columns), the difference of medians between privileged and unprivileged is given both before and after simulation. Refer to the bottom two rows to see by how much the median increased for each group.



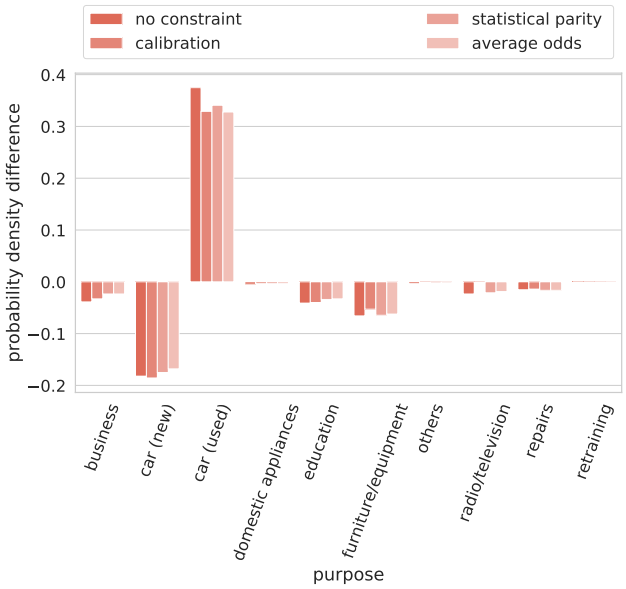
(a) Feature distribution (CDF) of *savings* for *unprivileged*



(b) Feature distribution (CDF) of *savings* for *privileged*

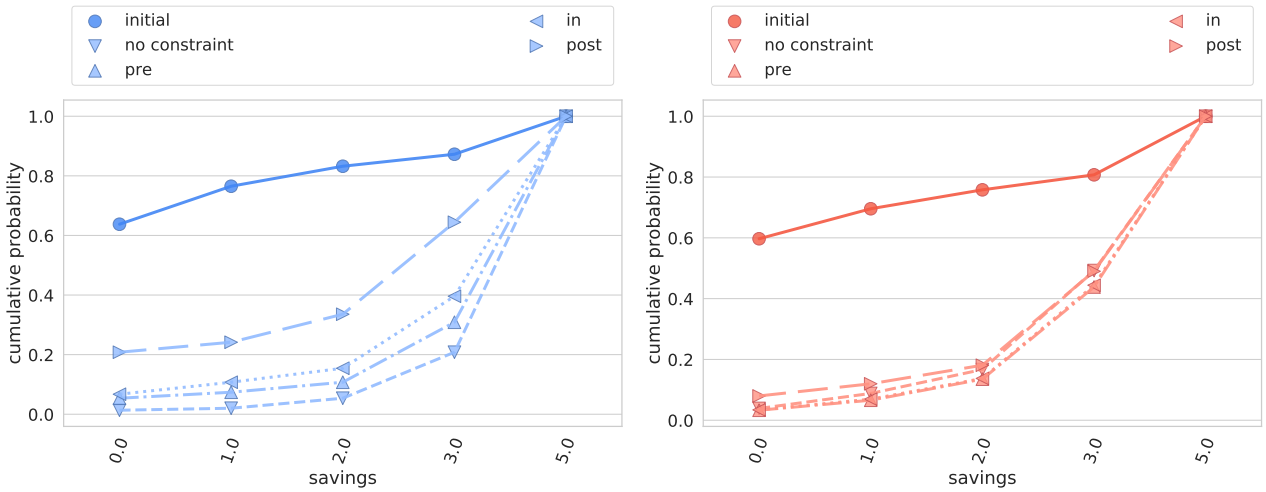


(c) Difference of feature distribution of *purpose* for *unprivileged*



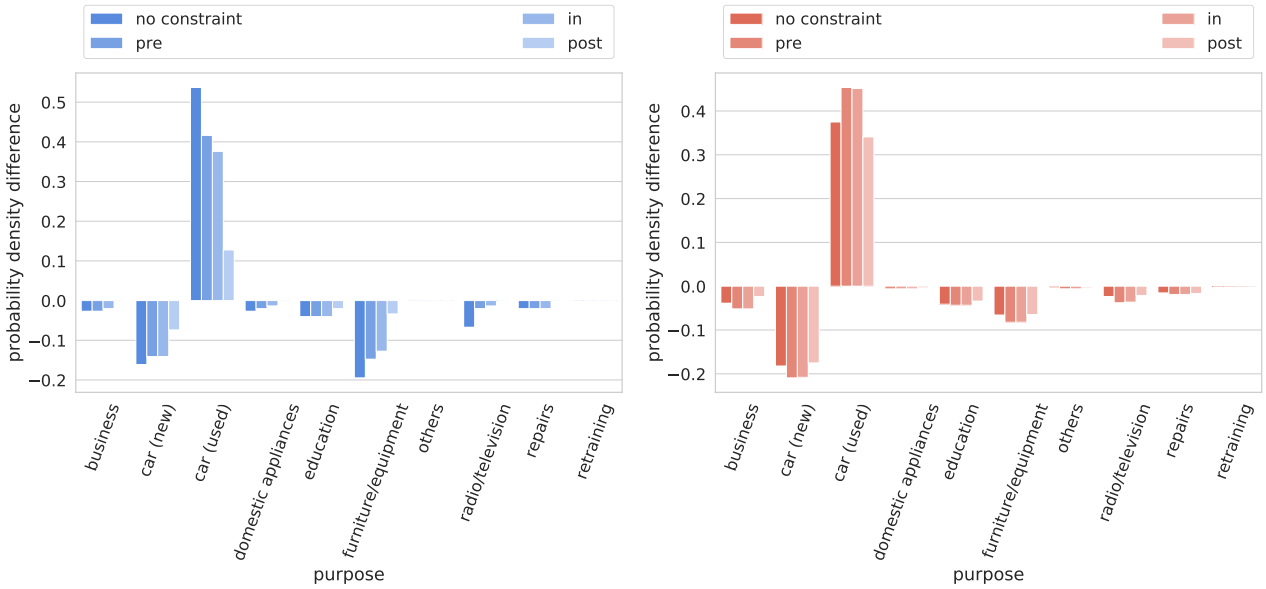
(d) Difference of feature distribution of *purpose* for *privileged*

Figure 3: While for the *privileged* group the imposed notion of fairness does not drastically change the impacted feature distribution (see Figure 3b and 3d), both *statistical parity* and *average odds* incentivize the *unprivileged* group to manipulate less (see Figure 3a and 3c)



(a) Feature distribution (CDF) of *savings* for *unprivileged*

(b) Feature distribution (CDF) of *savings* for *privileged*



(c) Difference of feature distribution of *purpose* for *unprivileged*

(d) Difference of feature distribution of *purpose* for *privileged*

Figure 4: The *privileged* group manipulates much the same, no matter the employed statistical parity intervention (see Figure 4b and 4d). The *unprivileged* group manipulates most if *no constraint* is enforced. With *in* and *pre* the *unprivileged* group manipulates marginally less and *post* leaves them *feature-wise much worse off than the other interventions* (see Figure 4a and 4c), indicating statistical parity might best be enforced using *pre-* or *in-processing* techniques.

5 Conclusion

We presented a theoretical framework to reason about the long term impact of different fairness constraints. Our experiments with this framework, although limited in scope, indicate that *statistical parity might best be enforced using pre- or in-processing techniques as post-processing techniques might create an undesirable local optima.*

We think that a *social learning approach* [Heidari *et al.*, 2019] for the agents best response may be a fruitful direction for further research.

References

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

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *Propublica*, 2016.

Anna Barry-Jester, Ben Casselman, and Dana Goldstein. The new science of sentencing. *The Marshall Project*, August 8 2015. Retrieved 4/28/2016.

Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, October 2018.

Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *arXiv preprint arXiv:1703.09207*, 2017.

Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. In *Data mining workshops, 2009. ICDMW'09. IEEE international conference on*, pages 13–18. IEEE, 2009.

L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 319–328. ACM, 2019.

Kim-Sau Chung. Role models and arguments for affirmative action. *American Economic Review*, 90(3):640–648, 2000.

Stephen Coate and Glenn Loury. Antidiscrimination enforcement and the problem of patronization. *The American Economic Review*, 83(2):92–98, 1993.

Stephen Coate and Glenn C Loury. Will affirmative-action policies eliminate negative stereotypes? *The American Economic Review*, pages 1220–1240, 1993.

Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. *arXiv preprint arXiv:1701.08230*, 2017.

Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 55–70. ACM, 2018.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226. ACM, 2012.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2015.

Sara Hajian, Josep Domingo-Ferrer, and Antoni Martinez-Balleste. Rule protection for indirect discrimination prevention in data mining. In *Modeling Decision for Artificial Intelligence*, pages 211–222. Springer, 2011.

Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, pages 3315–3323, 2016.

Robert David Hart. If you're not a white male, artificial intelligence's use in healthcare could be dangerous. *Quartz*, July 2017.

Hoda Heidari, Vedant Nanda, and Krishna P Gummadi. On the long-term impact of algorithmic decision policies: Effort unfairness and feature segregation through social learning. *arXiv preprint arXiv:1903.01209*, 2019.

Lily Hu and Yiling Chen. A short-term intervention for long-term fairness in the labor market. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1389–1398. International World Wide Web Conferences Steering Committee, 2018.

Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the 2nd ACM Conference on Fairness, Accountability, and Transparency*, 2019.

Faisal Kamiran and Toon Calders. Classifying without discriminating. In *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*, pages 1–6. IEEE, 2009.

Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ICDM '12, pages 924–929, Washington, DC, USA, 2012. IEEE Computer Society.

Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 924–929. IEEE, 2012.

Sampath Kannan, Aaron Roth, and Juba Ziani. Downstream effects of affirmative action. In *Proceedings of the 2nd ACM Conference on Fairness, Accountability, and Transparency*, 2019.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

Lydia T. Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. *arXiv preprint arXiv:1803.04383*, 2018.

Clair Miller. Can an algorithm hire better than a human? *The New York Times*, June 25 2015. Retrieved 4/28/2016.

Smitha Milli, John Miller, Anca D Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the 2nd ACM Conference on Fairness, Accountability, and Transparency*, 2019.

Hussein Mouzannar, Mesrob I Ohannessian, and Nathan Srebro. From fair decision making to social equality. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 359–368. ACM, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Kevin Petrasic, Benjamin Saul, James Greig, and Matthew Bornfreund. Algorithms and bias: What lenders need to know. *White & Case*, 2017.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems*, pages 5684–5693, 2017.

Cynthia Rudin. Predictive policing using machine learning to detect patterns of crime. *Wired Magazine*, August 2013. Retrieved 4/28/2016.

Latanya Sweeney. Discrimination in online ad delivery. *Queue*, 11(3):10, 2013.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness constraints: Mechanisms for fair classification. *arXiv preprint arXiv:1507.05259*, 2017.