Enforcing Delayed-Impact Fairness Guarantees

Aline Weber*
Manning College of Information and Computer Sciences
University of Massachusetts
Amherst, MA 01002
alineweber@cs.umass.edu

Blossom Metevier*
Manning College of Information and Computer Sciences
University of Massachusetts
Amherst, MA 01002
bmetevier@cs.umass.edu

Yuriy Brun
Manning College of Information and Computer Sciences
University of Massachusetts
Amherst, MA 01002
brun@cs.umass.edu

Philip S. Thomas
Manning College of Information and Computer Sciences
University of Massachusetts
Amherst, MA 01002
pthomas@cs.umass.edu

Bruno Castro da Silva
Manning College of Information and Computer Sciences
University of Massachusetts
Amherst, MA 01002
bsilva@cs.umass.edu

Abstract

Recent research has shown that seemingly fair machine learning models, when used to inform decisions that have an impact on peoples' lives or well-being (e.g., applications involving education, employment, and lending), can inadvertently increase social inequality in the long term. This is because prior fairness-aware algorithms only consider static fairness constraints, such as equal opportunity or demographic parity. However, enforcing constraints of this type may result in models that have negative delayed impact on disadvantaged individuals and communities. We introduce ELF (Enforcing Long-term Fairness), the first algorithm that provides high-confidence fairness guarantees in terms of delayed impact, using importance sampling techniques similar to those in the offline reinforcement learning literature. We prove that ELF will not return an unfair solution with probability greater than a user-specified tolerance. Furthermore, we show (under mild assumptions) that given sufficient training data, ELF is able to find and return a fair solution if one exists. We show experimentally that ELF can successfully mitigate long-term unfairness.

Keywords: Fair machine learning, delayed impact, fair classification

*Equal contribution.

1 Introduction

The use of machine learning for high-stakes applications such as lending, hiring, and criminal sentencing has the potential to harm historically disadvantaged communities [5, 3, 2]. For example, software meant to guide bank decisions in lending has been shown to exhibit racial bias [2]. Consequently, extensive research has been devoted to designing algorithms that promote fairness and ameliorate concerns of bias and discrimination for socially impactful applications. The bulk of this research has focused on the classification setting, in which *static* fairness definitions, i.e., fairness definitions that rely on statistical metrics such as true and false positive rates, are studied. However, it has been shown that model decisions that appear fair with respect to static fairness measures can nevertheless negatively affect the community they aim to protect in the long-term. For example, consider a bank lending setting, where the repayment predictions influence lending decisions and can change the future financial stability of different groups (e.g. by not getting a loan, the financial stability of a person may decay drastically). When a subset of the population is disadvantaged, instead of maximizing profit, the bank may want (or be required by law) to maximize profit subject to a fairness constraint that considers the delayed impact of model predictions in terms of the borrowers' future financial stability. Work that enforces long-term, or *delayed-impact* (DI) constraints when the relationship between predictions and DI is known has been proposed [7], but designing algorithms that mitigate negative delayed impact when the relationship between predictions and DI is not known has remained an open problem.

In this paper, we develop the first classification algorithm that can ensure with high probability that the classifiers it learns are fair with respect to delayed impact when the relationship between predictions and DI is not known *a priori*. To accomplish this, we simultaneously formulate the fair classification problem as both a classification and reinforcement learning problem—classification for optimizing the primary objective (a measure of classification loss) and reinforcement learning when considering DI. Specifically, we use importance sampling techniques similar to those in the offline reinforcement learning literature [8], and make use of confidence intervals for the mean [9] to derive a method for computing high probability bounds on DI fairness.

2 Problem Statement

We now formalize the problem of classification with delayed-impact fairness guarantees. As in the standard classification setting, a dataset consists of n data points, the i^{th} of which contains X_i, a feature vector describing a person, and a label Y_i. Each data point also contains a set of *sensitive attributes*, such as race and gender. Though our algorithm works with an arbitrary number of such attributes, for brevity our notation uses a single attribute, T_i. We assume that each data point also contains a prediction $\widehat{Y}_i^\beta$ made by a stochastic model β. We call β the *behavior model*, defined as $\beta(x, \hat{y}) \coloneqq \Pr(\widehat{Y}_i^\beta{=}\hat{y}|X_i{=}x)$. The predictions made by a model deployed in the real-world can have long-term, or delayed, impact. For example, by influencing who gets a loan, a model's predictions can affect applicants' long-term net worth. Formally, let I_i^β be a measure of the *delayed impact* resulting from deploying β for the person described by the i^{th} data point. We assume that larger values of I_i^β correspond to better delayed impact. We append I_i^β to each data point, and thus define the dataset to be a sequence of n independent and identically distributed (i.i.d.) data points $D \coloneqq \{(X_i, Y_i, T_i, \widehat{Y}_i^\beta, I_i^\beta)\}_{i=1}^n$. For notational clarity, when referring to an arbitrary data point, we write $X, Y, T, \widehat{Y}^\beta$ and I^β without subscripts to denote $X_i, Y_i, T_i, \widehat{Y}_i^\beta$ and I_i^β, respectively.

Given a dataset D, the goal is to construct a classification algorithm that takes as input D and outputs a new model π_θ that is as accurate as possible while enforcing constraints on delayed impact. This new model π_θ is of the form $\pi_\theta(x, \hat{y}) \coloneqq \Pr(\widehat{Y}^{\pi_\theta}{=}\hat{y}|X{=}x)$, where π_θ is parameterized by a vector $\theta \in \Theta$, for some feasible set Θ, and where $\widehat{Y}^{\pi_\theta}$ is the prediction made by π_θ given X. Like I_i^β, let $I_i^{\pi_\theta}$ be the delayed impact if the model outputs the prediction $\widehat{Y}_i^{\pi_\theta}$.

We model the setting in which a classification model's prediction depends only on the feature vector X, formalized by the assumption that for all x, t, y, and $\hat{y}$, $\Pr(\widehat{Y}^{\pi_\theta}{=}\hat{y}|X{=}x, Y{=}y, T{=}t) = \Pr(\widehat{Y}^{\pi_\theta}{=}\hat{y}|X{=}x)$. We also assume that regardless of the model used to make predictions, the distribution of delayed impact given a prediction remains the same: $\forall x, y, t, \hat{y}, i$, $\Pr(I^\beta{=}i|X{=}x, Y{=}y, T{=}t, \widehat{Y}^\beta{=}\hat{y}) = \Pr(I^{\pi_\theta}{=}i|X{=}x, Y{=}y, T{=}t, \widehat{Y}^{\pi_\theta}{=}\hat{y})$.

Our problem setting can alternatively be described from the reinforcement learning perspective, where feature vectors are the states of a Markov decision process (specifically, a contextual bandit), predictions are the actions taken by an agent, and DI is the reward received after the agent takes an action (makes a prediction) given a state (feature vector). From this perspective, the latter assumption asserts that regardless of the strategy (model) used to choose actions, the distribution of rewards given an action remains the same.

We consider k *delayed-impact objectives* $g_j : \Theta \to \mathbb{R}, j \in \{1, ..., k\}$ that take as input a parameterized model θ and return a real-valued measurement of fairness in terms of delayed impact. We adopt the convention that $g_j(\theta) \leq 0$ iff θ causes behavior that is fair with respect to delayed impact, and $g_j(\theta) > 0$ otherwise. To simplify notation, we assume there

exists only a single DI objective (i.e., $k = 1$). We focus on the case in which each DI objective is based on a conditional expected value, having the form $g(\theta) \coloneqq \tau - \mathbf{E}[I^{\pi_\theta}|c(X,Y,T)]$, where $\tau \in \mathbb{R}$ is a tolerance, and $c(X,Y,T)$ is a Boolean conditional relevant to defining the objective. Consider an example in which a bank determines whether to provide a loan to an applicant. We assume that each individual in the population of loan applicants is associated with type A or B, e.g., A and B can represent different genders. The bank is interested in enforcing a fairness definition that protects type A applicants. Specifically, the bank would like to ensure that, for a model π_θ being considered, the future financial status of applicants of type A impacted by π_θ's lending predictions does not decline relative to those induced by the previously deployed model, β. In this case, I^{π_θ} is the financial status of an applicant t months after the loan application and $c(X,Y,T)$ is the Boolean event that an applicant is of type A. Lastly, τ could represent a threshold on which the bank would like to improve, e.g., the average financial status of applicants in the disadvantaged group given the historical data collected using β. The bank is therefore interested in enforcing the following DI objective: $\mathbf{E}[I^{\pi_\theta}|T = A] \geq \tau$. Then, defining $g(\theta) = \tau - \mathbf{E}[I^{\pi_\theta}|T = A]$ ensures that $g(\theta) \leq 0$ iff the new model π_θ satisfies the DI objective. Note that an additional constraint of the same form can be added to protect group B.

Algorithmic properties of interest. We would like to ensure that $g(\theta) \leq 0$, where θ is the model returned by a classification algorithm. However this is often not possible, as it requires highly accurate assumptions of prior knowledge about how predictions influence delayed impact. Instead, we aim to create an algorithm that uses data to reason about its confidence that $g(\theta) \leq 0$. That is, we desire a classification algorithm, a, where $a(D) \in \Theta$ is the solution provided by the algorithm when given dataset D as input, that satisfies DI constraints of the form

$$\Pr(g(a(D)) \leq 0) \geq 1 - \delta, \tag{1}$$

where $\delta \in (0,1)$ limits the admissible probability that the algorithm returns a model that is unfair with respect to the DI objective. Algorithms that satisfy (1) are called Seldonian [10]. In practice, there might be constraints that are impossible to enforce [6] or the amount of data may be insufficient to ensure fairness with high confidence. In such cases, instead of returning a solution the algorithm does not trust, the algorithm should return "No Solution Found" (NSF). Let $\texttt{NSF} \in \Theta$ and $g(\texttt{NSF}) = 0$, indicating that it is always fair for the algorithm to say "I'm unable to ensure fairness with the required confidence."

3 Methods for Enforcing Delayed Impact

The distribution of delayed impacts in D is a result of using the model β to make predictions. However, we are interested in evaluating the DI of a different model, π_θ. This presents a challenging problem: given data that includes the DI when a model β was used to make predictions, how can we estimate the DI if π_θ were used instead?

We solve this problem using techniques from the reinforcement learning literature called *off-policy evaluation* methods—methods that use data from running one *policy* (decision-making model) to predict what would happen (in the long-term) if a different policy were used to make decisions. Specifically, we use an off-policy evaluation method called *importance sampling* [8] to obtain a new random variable $\hat{I}^{\pi_\theta}$, constructed using data from β, such that $\mathbf{E}[\hat{I}^{\pi_\theta}|c(X,Y,T)] = \mathbf{E}[I^{\pi_\theta}|c(X,Y,T)]$. For each data point, the importance sampling estimator, $\hat{I}^{\pi_\theta}$, weights the observed delayed impacts I^β based on how likely the prediction $\widehat{Y}^\beta$ is under π_θ. If π_θ would make the label $\widehat{Y}^\beta$ more likely, then I^β is given a larger weight (at least one), and if π_θ would make $\widehat{Y}^\beta$ less likely, then I^{π_θ} is given a smaller weight (positive, but less than one). Formally, the importance sampling estimator is $\hat{I}^{\pi_\theta} = \pi_\theta(X,\widehat{Y}^\beta)\big(\beta(X,\widehat{Y}^\beta)\big)^{-1} I^\beta$, where the term $\pi_\theta(X,\widehat{Y}^\beta)/\beta(X,\widehat{Y}^\beta)$ is called the *importance weight*. This particular weighting scheme is chosen to ensure that $\hat{I}^{\pi_\theta}$ is an unbiased estimator of I^{π_θ}, which can be proven under the assumption that the model π_θ can only select labels for which there is *some* probability of the behavior model selecting.

Bounds on delayed impact. Given unbiased estimates of I^{π_θ}, computed according to the scheme discussed above, we can construct unbiased estimates of $g(\theta)$ by subtracting each estimate of I^{π_θ} from τ, the user-defined tolerance. We now discuss how to use these estimates of $g(\theta)$, along with confidence intervals for the mean, to derive high confidence upper bounds on $g(\theta)$. Given a vector of m i.i.d. samples $(Z_i)_{i=1}^m$ of a random variable Z, let $\bar{Z} = \frac{1}{m}\sum_{i=1}^m Z_i$ be the sample mean, let $\sigma(Z_1,...,Z_m) = \sqrt{\frac{1}{m-1}\sum_{i=1}^m (Z_i - \bar{Z})^2}$ be the sample standard deviation (with Bessel's correction), and let $\delta \in (0,1)$ be a confidence level. From Student [9], we have the property that if $\sum_{i=1}^m Z_i$ is normally distributed, then $\Pr\left(\mathbf{E}[Z_i] \geq \bar{Z} - \frac{\sigma(Z_1,\ldots,Z_m)}{\sqrt{m}} t_{1-\delta,m-1}\right) \geq 1 - \delta$, where $t_{1-\delta,m-1}$ is the $1-\delta$ quantile of the Student's t distribution with $m-1$ degrees of freedom. We can use this property to obtain a high-confidence upper bound for the mean of Z : $U_{\texttt{ttest}}(Z_1,\ldots,Z_m) = \bar{Z} + \frac{\sigma(Z_1,\ldots,Z_m)}{\sqrt{m}} t_{1-\delta,m-1}$.

Let $\hat{g}$ be a vector of i.i.d. and unbiased estimates of $g(\theta)$ such that the sample mean of $\hat{g}$ is normally distributed. These estimates can be provided to $U_{\texttt{ttest}}$ to derive a high-confidence upper bound on $g(\theta)$: $\Pr(\tau - \mathbf{E}[\hat{I}^{\pi_\theta}|c(X,Y,T)] \leq U_{\texttt{ttest}}(\hat{g})) \geq 1 - \delta$.

Complete algorithm. Our algorithm (Algorithm 1) has three main steps. In the first step, the dataset D is divided into two datasets, D_c and D_f (line 1). In the second step (line 3), which we refer to as *candidate selection*, D_c is used to find and train a model, called the *candidate solution*, θ_c. The last step (lines 4–9) is the *fairness test*, in which D_f is used to compute a $(1-\delta)$-confidence upper bound on $g(\theta_c)$ and determine whether `NSF` or the candidate solution should be returned.

In particular, in the fairness test, unbiased estimates of $g(\theta_c)$ are calculated using the importance sampling method described previously (lines 4–7). These estimates are used to calculate a high-confidence upper bound, U, on $g(\theta_c)$ using Student's t-test (line 9). Finally, if U is below 0, then the solution θ_c is returned. If not, the algorithm returns `NSF`. In candidate selection, a similar strategy is used to calculate the cost of a potential solution θ. Again, unbiased estimates of $g(\theta)$ are calculated (Algorithm 2 lines 2–6), this time using D_c. Instead of calculating a high confidence upper bound on $g(\theta)$ using Student's t-test, we calculate an *inflated* upper bound (Algorithm 2 lines 7–8). Our choice to inflate the confidence interval is empirically driven and was first proposed for other Seldonian algorithms [10]. Finally, if the inflated upper bound is higher than a small negative constant $(-\xi/4)$, the cost associated with the loss of θ, $\hat{\ell}(\theta, D_c)$, is returned. Otherwise, the cost of θ is defined as the sum of the inflated upper bound and the maximum loss that can be obtained using D_c (Algorithm 2 lines 9–10). This discourages candidate selection from returning models unlikely to pass the fairness test.

Algorithm 1 `ELF`$(D, c, \delta, \tau, \beta)$

1: $D_c, D_f \leftarrow$ `partition`(D)
2: $n_{D_f} =$ `length`(D_f); $\quad \hat{g} \leftarrow \langle\,\rangle$
3: $\theta_c \leftarrow \arg\min_{\theta\in\Theta}$ `cost`$(\theta, D_c, c, \delta, \tau, \beta, n_{D_f})$
4: **for** $i \in \{1, ..., n\}$ **do**
5: **if** $c(X_i, Y_i, T_i)$ is `True` **then**
6: $\hat{g}$.append$\Big(\tau - \frac{\pi_{\theta_c}(X_i, \widehat{Y}_i^{\beta})}{\beta(X_i, \widehat{Y}_i^{\beta})} I_i^{\beta}\Big)$
7: **end if**
8: **end for**
9: **if** $U_{\texttt{ttest}}(\hat{g}) \geq 0$ **then return** `NSF` **else return** θ_c

4 Empirical Evaluation

To empirically evaluate our method, we consider a classifier tasked with making predictions about people in the United States foster care system; for example, whether youth currently in foster care are likely to get a job in the near future. These predictions may have a delayed impact on the person's life if, for instance, they influence whether that person receives additional financial aid. Here the goal is to ensure that a trained classifier is fair with respect to delayed impact when considering race. Our experiments use two data sources from the National Data Archive on Child Abuse and Neglect [4]: *(i)* the Adopting and Foster Care Analysis and Reporting System—a dataset containing demographic and foster care-related information about youth; and *(ii)* the National Youth in Transition Database (Services and Outcomes)—a dataset containing information about the well-being, financial, and educational status of youth over time and during their transition from foster care to independent adulthood. We wish to guarantee with high probability that the DI caused by a new classifier, π_θ, is better than the DI resulting from the currently-deployed classifier, β. This guarantee should hold simultaneously for both races: White (instances where $T = 0$) and Black (instances where $T = 1$). In the following experiments, the confidence levels δ_0 and δ_1, associated with these objectives, are both set to 0.1.

Preventing Delayed-Impact Unfairness. We first evaluate whether `ELF` can prevent DI unfairness with high probability, and whether existing algorithms fail. We compare `ELF` with a fairness-unaware algorithm (logistic regression (LR)) and three state-of-the-art fairness-aware algorithms: *(i)* Fairlearn [1], *(ii)* Fairness Constraints [11], and *(iii)* quasi-Seldonian algorithms (QSA) [10] designed to enforce static fairness constraints. We consider five static fairness constraints: demographic parity (DP), equalized odds (EqOdds), disparate impact (DisImp), equal opportunity (EqOpp), and predictive equality (PE).

Algorithm 2 `cost`$(\theta, D_c, c, \delta, \tau, \beta, n_{D_f})$

1: $\hat{g} \leftarrow \langle\,\rangle$
2: **for** $i \in \{1, ..., m\}$ **do**
3: **if** $c(X_i, Y_i, T_i)$ is `True` **then**
4: $\hat{g}$.append$\Big(\tau - \frac{\pi_{\theta}(X_i, \widehat{Y}_i^{\beta})}{\beta(X_i, \widehat{Y}_i^{\beta})} I_i^{\beta}\Big)$
5: **end if**
6: **end for**
7: Let $\lambda = 2$; $\quad n_{\hat{g}} =$ `length`$(\hat{g})$
8: $U^+ = \frac{1}{n_{\hat{g}}}\left(\sum_{\iota=1}^{n_{\hat{g}}} \hat{g}_\iota\right) + \lambda \frac{\sigma(\hat{g})}{\sqrt{n_{D_f}}} t_{1-\delta, n_{D_f}-1}$
9: $\ell_{\max} = \max_{\theta'\in\Theta} \hat{\ell}(\theta', D_c)$
10: **if** $U^+ \leq -\frac{\xi}{4}$ **return** $\hat{\ell}(\theta, D_c)$ **else return** $(\ell_{\max} + U^+)$

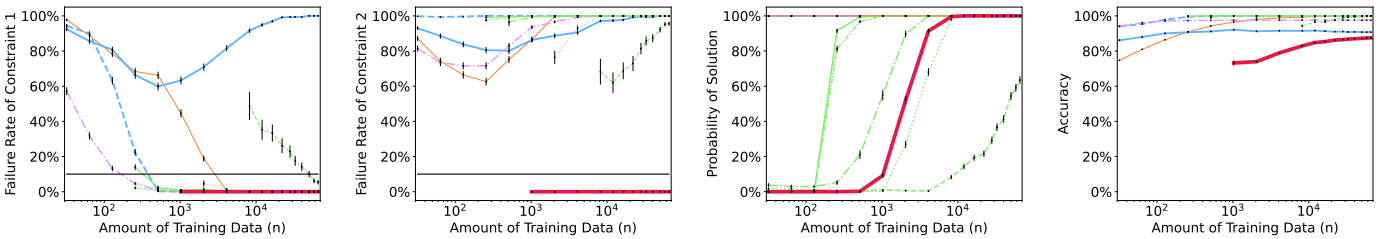
In this comparison, we investigate how often each fairness-aware algorithm returns an unfair model (with respect to the DI constraints) as a function of the amount of training data. We refer to the probability that an algorithm returns an unfair model as its *failure rate*. To measure the failure rate, we compute how often the classifiers returned by each algorithm are unfair when evaluated on a significantly larger dataset, to which the algorithms do not have access during training time. Figures 1a and 1b present the failure rate of each algorithm as a function of the amount of available training data. We computed all failure rates and corresponding standard errors over 500 trials. Notice that the solutions returned by `ELF` are *always fair* with respect to the DI constraints.[1] Existing methods that enforce static fairness criteria, by contrast, either *(i) always fail* to satisfy both

[1] `ELF` does not return solutions if trained with $n < 1{,}000$ data points because it cannot ensure DI fairness with high confidence.

DI constraints, independently of the amount of training data; or *(ii) always fail* to satisfy one of the DI constraints—the one related to delayed impact on Black youth.

The Cost of Ensuring Delayed-Impact Fairness. The previous analyses show that `ELF` is capable of satisfying DI constraints with high probability, but this often comes at a cost. First, there may be a trade-off between the amount of training data and the confidence that a fair solution has been identified. Recall that some algorithms (including ours) may not return a solution if they cannot ensure fairness with high confidence. Therefore, we study how often each algorithm identifies and returns a candidate solution as a function of n. Figure 1c shows that as the amount of training data increases, the probability of `ELF` returning solutions increases rapidly. Although some competing techniques always return solutions, or may require less training data than `ELF`, these solutions never satisfy both DI constraints.

Secondly, there may be a trade-off between satisfying fairness constraints and optimizing accuracy. Figure 1d presents the accuracy of classifiers returned by different algorithms as a function of n. Even though there is an accuracy gap of approximately 10% in the limit, `ELF` *always* returns fair solutions, while other methods fail to satisfy at least one DI constraint. *While there is a cost to enforcing DI constraints, `ELF` succeeds in its main objectives: to ensure DI fairness with high probability, without requiring unreasonable amounts of data, and with no significant loss of accuracy.*



(a) Failure rate 1. (b) Failure rate 2. (c) Prob. of returning solution. (d) Accuracy.

Figure 1: Algorithms' failure rates with respect to the DI constraints associated with White people *(a)* and Black people *(b)*, as a function of n. The black horizontal lines indicate the maximum admissible probability of unfairness, $\delta_0 = \delta_1 = 10\%$. In *(c)* we show the probability that algorithms (subject to different fairness constraints) return a solution as a function of n. Finally, in *(d)* we show the accuracy of the solutions returned by algorithms (subject to different fairness constraints) as a function of n. All plots use the following legend: ELF LR QSA with DP QSA with EqOdds QSA with EqOpp QSA with PE QSA with DisImp Fairlearn with DP Fairlearn with EqOdds Fairness Constraints.

References

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

[2] Robert Bartlett, Adair Morse, Richard Stanton, and Nancy Wallace. Consumer-lending discrimination in the fintech era. *Journal of Financial Economics*, 2021.

[3] Joseph Blass. Algorithmic advertising discrimination. *Northwestern University Law Review*, 114:415–468, 2019.

[4] Children's Bureau, Administration on Children, Youth and Families. National Data Archive on Child Abuse and Neglect (NDACAN). 2021.

[5] Alexandre Flage. Ethnic and gender discrimination in the rental housing market: Evidence from a meta-analysis of correspondence tests, 2006–2017. *Journal of Housing Economics*, 41:251–273, 2018.

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

[7] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In *International Conference on Machine Learning*, pages 3150–3158. PMLR, 2018.

[8] D. Precup, R. S. Sutton, and S. Dasgupta. Off-policy temporal-difference learning with function approximation. In *Proceedings of the 18th International Conference on Machine Learning*, pages 417–424, 2001.

[9] Student. The probable error of a mean. *Biometrika*, pages 1–25, 1908.

[10] Philip S Thomas, Bruno Castro da Silva, Andrew G Barto, Stephen Giguere, Yuriy Brun, and Emma Brunskill. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468):999–1004, 2019.

[11] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR, 2017.