Augmenting Fairness With Welfare: A Framework for Algorithmic Justice

Sílvia Casacuberta[1], Isaac Robinson[1] and Connor Wagaman[2]

[1]Harvard University, Cambridge, MA, USA
[2]Boston University, Boston, MA, USA

Abstract

In this paper, we propose a utility-based framework for *algorithmic justice*. Justice and fairness are often discussed alongside one another in philosophical contexts, and, though related, they remain distinct concepts. In computer science, some notions of fairness rely on an *ex ante* interpretation: an algorithm is fair if similar individuals experience similar outcomes in expectation. We argue that the *ex post* behavior of an algorithm is also important and that an algorithm can be fair but, if the *ex post* output is poorly aligned with the input, unjust.

We explore the distinction between justice and fairness in algorithmic contexts and motivate the need for a formal notion of algorithmic justice by presenting several examples of algorithms that are algorithmically fair but philosophically unjust. We then propose a formal mathematical definition for algorithmic justice that is rooted in well-established philosophical conceptions of justice, and which leverages social welfare functions, a tool from welfare economics which provides a relative measurement for how different mappings from individuals to outcomes can affect society. Following the presentation of this definition, we develop several results relating fairness and justice, including impossibility results for fairness and justice, and a structure for creating algorithms that fulfill both fairness and justice.

Keywords

algorithmic fairness, social welfare, utility functions, justice

1. Introduction

Algorithms inform decisions which can greatly impact lives, predicting everything from the chance of a borrower repaying a loan [1] to the probability of a criminal re-offending [2]. The prevalence of algorithms in everyday life, increased by the widespread adoption of machine learning, means that analyzing algorithms' impacts – in particular, the ways in which an algorithm positively favors or negatively discriminates against groups and individuals – is imperative. A formal framework for algorithmic fairness was first presented by Dwork et al. in 2012 [3]. Informally, the definition of (individual) algorithmic fairness presented in [3] requires that similar individuals are treated similarly; more formally, it requires that similar individuals are assigned outcomes drawn from similar probability distributions.

While algorithmic fairness requires that similar individuals are assigned outcomes drawn from similar probability distributions, it offers no restrictions on whether individuals are assigned to appropriate outcomes. The definition ensures that similar individuals are treated similarly *ex ante*, but it imposes no restrictions on the *ex post* behavior of algorithms. This stands in contrast to our existing legal system where there is little concern as to whether justice was *expected* to be served but rather the emphasis is on whether justice *was* served – that is, they care whether the actual outcome

EWAF'23: European Workshop on Algorithmic Fairness, June 7–9, 2023, Winterthur, Switzerland
scasacubertapuig@college.harvard.edu (S. Casacuberta); isaac_robinson@college.harvard.edu (I. Robinson); wagaman@bu.edu (C. Wagaman)
0000-0001-5684-4585 (S. Casacuberta); 0000-0003-2459-3886 (C. Wagaman)

of the trial follows logically from the proceedings and the evidence. Imposing restrictions on the *ex post* behavior of an algorithm, then, is useful. In this paper, we describe a set of criteria that can be imposed on the *ex post* behavior of an algorithm to move towards this same notion of *justice*.

In this paper, we explore the limitations of fairness, with a focus on individual fairness,[1] and describe ways in which fair algorithms can lead to *unjust* outcomes. Broadly, we study how to establish appropriate mappings from individuals to outcomes. Inspired by John Rawls' characterization of justice [4], we identify two varieties of intuitively unjust mappings from individuals' scores to outcomes: (1) *arbitrary* assignments to outcomes, and (2) *unreasonable* assignments to outcomes. We present several motivating examples of unjust algorithms in Section 2. In Section 4 we formalize the notions of allocation rule, utility function, and social welfare function (these concepts are used in the definition of algorithmic justice that we introduce in Section 3). We summarize the use of social welfare functions in the social choice and economics literature and present several concrete examples of social welfare functions, such as inequality metrics.

In Section 3, we introduce our definition of algorithmic justice (Definition 3.2). We begin by describing the setting to which our definition applies, and we also establish a connection between preventing arbitrary assignments to outcomes and the setting of differential privacy. We mathematically formalize the notions of arbitrary assignments and unreasonable assignments that we described through examples in Section 2. We address goal (1) of avoiding arbitrary assignments by imposing an upper bound on the standard deviation of the probability distribution from which scores for individuals are drawn. We address goal (2) of avoiding unreasonable outcomes by introducing an allocation rule and a social welfare function. This is motivated by the widespread use in social choice theory and welfare economics of *utility functions* to inform policy-making and capture notions of fairness [5].

In Section 5 we explain why some seemingly straight-forward ways to fix the issues presented in our motivating examples do not work, further demonstrating the value of our framework. In Section 6 we investigate the relationship between justice and fairness, provide several impossibility results (Theorems 6.1, 6.2, and 6.3), and demonstrate how our proposed framework is able to resolve the issues presented in our motivating examples. In Section 7 we describe more explicitly how our definition interacts with other notions of fairness. In Section 8, we conclude with some cautionary notes and an overview of our framework.

2. Motivating Examples

Motivated by the distinction between fairness and justice described in the introduction, in this section we present several examples of algorithms that are individually fair but which result in unjust assignments to outcomes. We give examples for both of the broad categories of injustice that we identify: assignments of individuals to *unreasonable* outcomes and assignments of individuals to (needlessly) *arbitrary* outcomes.

Before proceeding with the examples, we present the definition of (individual) fairness from [3].

Definition 2.1 (Lipschitz Mapping [3])**.** Let $\mathcal{X}$ be the set of individuals, $\mathcal{O}$ be the set of outcomes, and $\Delta(\mathcal{O})$ be the set of probability distributions over $\mathcal{O}$, $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be some metric[2] on pairs of individuals, and $D : \Delta(\mathcal{O}) \times \Delta(\mathcal{O}) \to \mathbb{R}$ be some metric on probability distributions.

[1] Throughout this paper, we will use the term "fairness" to refer to "individual fairness" as defined in Dwork et al. [3].

[2] We, and the authors of [3], use the term "metric" although some of the metrics are actually pseudometrics in that they allow distances of 0 for unequal elements and dist; and we may also allow distances of ∞. The motivation for this is that viewing distinct elements as equal (distance 0) or as being infinitely far (distance ∞) may make sense in the context of assigning outcomes.

A mapping $\mathcal{M} : \mathcal{X} \to \Delta(\mathcal{O})$ satisfies the (D, d)-Lipschitz property if for every $x, y \in \mathcal{X}$, we have

$$D(\mathcal{M}(x), \mathcal{M}(y)) \leq d(x, y).$$

Definition 2.2 ((Individual) Fairness [3])**.** Let $f : \mathcal{X} \to \mathcal{O}$ be a function mapping individuals to outcomes, and let $\mathcal{M}_f(\cdot)$ be the probability distribution from which the output of $f(\cdot)$ is drawn. If the mapping $\mathcal{M}_f$ satisfies Definition 2.1 for (D, d), we say that f is "(D, d)-individually fair" (usually omitting the adjective individually, and omitting (D, d) when the choice of (D, d) is clear).

2.1. Unreasonable Assignments to Outcomes

Fair algorithms can be unjust if they assign outcomes in unreasonable ways. We provide two examples that suggest *unreasonable* assignments in the context of allocations.

Example 2.3 (Unjust University Admissions)**.** Suppose a community with n university applicants has a single university with $m < n/2$ admission spots, and that all of these spots must be filled. (Assume that this community exists in isolation. Because there is only one university, the community does not care about factors like yield rate. Additionally, assume that the community has decided that the spots in the university should be given to the students who are most likely to succeed.) Consider an individually fair and accurate scoring function $f : \mathcal{X} \to [0, 1]$, where higher scores indicate a higher probability of success in university; and consider an admission recommendation function $g : [0, 1] \to \mathcal{C}$, where $\mathcal{C}$ is the set of potential recommendations (e.g., $\mathcal{C} = \{\text{admit, reject, waitlist}\}$). For simplicity, assume that a score $s \in \mathcal{S}$ is (a prediction of) the probability of success in college; in this example, we have $\mathcal{S} = [0, 1]$.

We have that the scoring function predicts success fairly (similar students receive similar predictions of success) and accurately. However, suppose the recommendation for whether to admit a student is done in a counter-intuitive, "flipped" manner: for a score $s \in \mathcal{S}$, let g map to "admit the student" with probability $(1 - s)$ and maps to "don't admit" with probability s.

The *social planner*, or in this case the community, would be expected to have a goal of giving spots in the university to the students who are most likely to succeed in college. However, the composition of our prediction algorithm and allocation rule $g \circ f$ does the opposite. The algorithm $g \circ f$ is still fair because similar students are mapped to outputs drawn from similar probability distributions. We argue, though, that $g \circ f$ is unjust because more-qualified students (i.e., students with higher probabilities of success) are less likely to be admitted than less-qualified students (i.e., students with lower probabilities of success).

Example 2.4 (Unjust Taxation)**.** In many European countries, the current system of taxation is often fair but unjust. For the most part, individuals with higher incomes pay higher taxes; however, because taxes on long-term capital gains are lower than taxes on ordinary income, individuals at the highest income levels can have a lower effective tax rate than the median household income.

Suppose that, in response to public discontent surrounding the current taxation system, some (malicious) wealthy individual offers to fund an effort to devise a fair taxation system. The community decides on a distance metric where people are close if they have similar abilities to pay taxes. Additionally, the scoring function f is fair, and it is made so that higher scores correspond to an increased ability to pay taxes. In short, the metric and the scoring function appear principled and reasonable.

The wealthy person then reveals the new, fair taxation algorithm g that maps scores to tax rates – and $g \circ f : \mathcal{X} \to \mathcal{C}$ is the same as the original taxation system. Individuals with similar abilities to pay taxes (similar incomes) pay similar tax rates; however, while average tax rates at first increase as scores increase, they eventually peak and then begin to decrease as scores increase. Everyone in the

society agreed on the distance metric between inputs, the scoring function f is fair, the algorithm g that determines consequences for scores is fair, and the overall algorithm $g \circ f$ is fair. However, the consequences are unreasonable, so we argue that the mapping is unjust.

2.2. Arbitrary Assignments to Outcomes

Fair algorithms can be also unjust if they assign outcomes in needlessly arbitrary ways. We provide an example illustrating how this issue can occur.

Example 2.5 (Unjust Prison Sentencing)**.** Suppose that the prison system of some country in the EU prefers to give harsh prison sentences, but the country's legislature has just passed a requirement that prison sentences be assigned such that, for every type of crime, the average prison sentence for the crime must be close to the recommended sentence for that type of crime. The prison system, though, decides to assign sentences according to someone's fairly-computed "crime score" (where higher scores are given for worse crimes) in the following way: with probability $1/2$, the individual is assigned to a sentence drawn from a distribution of sentences centered at "$0.05 \times$ recommended sentence"; with probability $1/2$, the individual is assigned to a sentence drawn from a distribution of sentences centered at "$1.95 \times$ recommended sentence". Assume that these distributions are scaled in such a way that they offer fairness for some pair of metrics.

This algorithm is ex-ante fair: similar individuals receive prison sentences drawn from similar probability distributions. However, similar individuals can end up assigned to wildly different severities of punishments – namely one that is centered at 0.05 times as harsh as recommended and the other that is centered at 1.95 times as harsh.

The injustice of this example is particularly clear in the following example: consider two twins, Twin 1 and Twin 2, who lived the exact same life and were accomplices in committing the exact same crime. With probability 1/2, then, Twin 1 and Twin 2 receive outcomes that are essentially drawn from different distributions with centers that are far from one another.

3. Algorithmic Justice

From the examples in Section 2, we see that algorithms which fulfill the individual algorithmic fairness notion can still produce unjust results. These unjust results arise due to the fact that the only restriction on outcomes for fair functions is that similar individuals map to outcomes drawn from similar probability distributions; as long as this condition is met, fairness will be fulfilled regardless of whether these outcomes align with any notion of what outcome the individual will actually receive. In this section, we describe philosophical requirements that should be met by systems claiming to be just, and we then provide a mathematical definition for algorithmic justice that ensures functions exhibit these features.

Relationship to Differential Privacy (DP). We hypothesize that the potential for fair algorithms to be unjust arises in part due to the close theoretical connection between individual fairness and differential privacy [6]. Privacy-preserving data analysis has two opposing players: honest data analysts want to be able to get accurate statistics about a dataset, while privacy-concerned individuals want their sensitive information to be protected from malicious data analysts. Data analysts have an inherent desire to get accurate answers, meaning that a function satisfying DP will often offer the best accuracy possible permitted by the privacy-loss parameter. In fairness, there is less of an obvious two-player relationship and, as we show in Section 2, there can be malicious motivations for drawing outputs from a distribution with higher standard deviation than necessary (Example 2.5) or offering

answers that are far from those that make intuitive sense with respect to the scores (Examples 2.3 and 2.4). Stating the problem in the context of DP, there is no clear adversarial incentive for providing answers that are less accurate than needed. In this section, we propose algorithmic justice both as a general notion for what justice means in an algorithmic context, and as a defense against allowing fair algorithms that adversarially assign individuals to outcomes.

3.1. The Setting

We use a setting similar to the one presented by Dwork et al. [3]. In this paper, we consider classifiers that map individuals to outcomes. We denote the set of individuals by $\mathcal{X}$, the set of outcomes by $\mathcal{O}$, and the set of probability distributions over outcomes $\mathcal{O}$ by $\Delta(\mathcal{O})$. In the simplest case, we can have the binary setting $\mathcal{O} = \{0, 1\}$. As in [3], we assume the existence of a metric on individuals $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

For the notion of individual fairness (IF), we consider *randomized* mappings $f : \mathcal{X} \to \mathcal{O}$ from individuals to outcomes. When the outcomes are scores in the range $[0, 1]$, we call $f : \mathcal{X} \to [0, 1]$ a *scoring function*. There are two key properties from individual fairness that we will employ in our study: *similar treatment of similar inputs* and *robustness of fairness to post-processing*.

Recall from Definition 2.2 that, for a function $f : \mathcal{X} \to \mathcal{O}$ mapping individuals to outcomes, if the probability distribution $\mathcal{M}_f : \mathcal{X} \to \Delta(\mathcal{O})$ from which the output of f is drawn satisfies Definition 2.1 for some (D, d), then we say that f is (D, d)-fair.

In this paper, we only consider the setting where the metric D is D_∞ (i.e., the *relative ℓ_∞ metric*), although extensions are certainly possible (e.g., to D_{TV}). We use the following definition of D_∞:

$$D_\infty(P, Q) = \sup_{a \in A} \log \Big(\max \Big\{ \frac{P(a)}{Q(a)}, \frac{Q(a)}{P(a)} \Big\} \Big).$$

In addition to using similar treatment of similar inputs, we employ the *post-processing* property of fair mappings from [3]: namely, if $f : \mathcal{X} \to \mathcal{O}$ is (D_∞, d)-fair (meaning the probability distribution $\mathcal{M}_f : \mathcal{X} \to \Delta(\mathcal{O})$ from which the output of f is drawn is a (D_∞, d)-Lipschitz mapping) and $g : A \to B$ is any possibly randomized function, then $\mathcal{M}_{g \circ f} : \mathcal{X} \to \Delta(B)$ is (D_∞, d)-Lipschitz, so $g \circ f$ is (D_∞, d)-fair. In words, this means that fairness is robust to post-processing.

3.2. Key Elements of Our Framework

Our framework is motivated by two key elements of justice that are captured in the following quotes from John Rawls' *A Theory of Justice* [4].

1. (No arbitrary assignments of individuals to outcomes.) "Institutions are just when no arbitrary distinctions are made between persons in the assigning of basic rights and duties..." [4, p. 5]. We capture this notion by enforcing an upper bound on the **standard deviation** of the probability distribution from which outcomes are drawn.

2. (No unreasonable assignments of individuals to outcomes.) The principles of justice "provide a way of assigning rights and duties in the basic institutions of society and they define the appropriate distribution of the benefits and burdens of social cooperation" [4, p. 4]. We capture this notion through the use of **social welfare functions**, a tool from welfare economics which provides a relative measurement for how assignments to outcomes affect society. We formally define and describe social welfare functions in Section 4.

Before providing a formal mathematical definition of algorithmic justice, we define the term informally in order to convey the intuition behind our definition.

Definition 3.1 (Algorithmic Justice (Informal))**.** An algorithm is just if (with high probability) an individual receives the most suitable outcome or an outcome that is close to the most suitable outcome.

Definition 3.2 (Algorithmic Justice (Formal))**.** Let $\mathcal{X}$ be the set of all size-n collections of individuals, and let $\mathcal{O}$ be the set of outcomes. Additionally, let $\mathcal{C} \subseteq \mathcal{O}^n$ be the (potentially resource-constrained) set of possible assignments of outcomes to all n individuals. A mapping $F : \mathcal{X} \to \mathcal{O}$, which can be decomposed into $F = g \circ f$ where $f : \mathcal{X} \to [0, 1]$ and $g : [0, 1] \to \mathcal{O}$, is *"$\alpha$-just with respect to social welfare function $\mathcal{W}$"*, where $\mathcal{W} : [0, 1] \times \mathcal{O} \to \mathbb{R}$, if:

1. For all $i \in [n]$, $\mathcal{M}_f(x_i)$ has standard deviation $\leq \alpha$.
2. For all $i \in [n]$, $g(s_i) = [\arg\max_{C \in \mathcal{C}} \mathcal{W}(S, C)]_i$.

Condition 1 provides an upper bound on the standard deviation of the distribution over scores, and it is meant to protect against the arbitrary assignment of individuals to outcomes. Condition 2 maximizes social welfare relative to constraints, and it is meant to prevent an unreasonable assignment of individuals to outcomes. We will be interested in comparing fairness with justice. In order to match both settings, we take advantage of the fact that fairness is robust to post-processing to visualize both terms together as shown in Fig. 1.

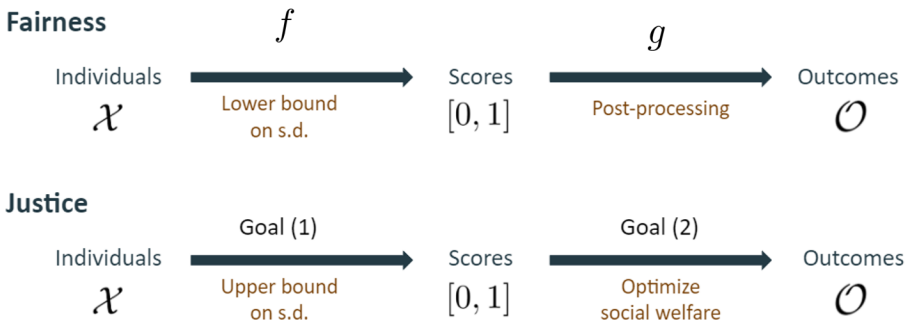

Figure 1: A diagram comparing fair algorithms and just algorithms. Goal (1) refers to item 1 of Definition 3.2; likewise, Goal (2) refers to item 2 of Definition 3.2. The abbreviation "s.d." stands for *standard deviation*.

Namely, we model the process of assigning individuals to outcomes in two steps. First, representations of individuals are fed into a potentially fair algorithm that produces score p_i. Those p_i values are then sorted into outcomes or results using some allocation rule. This breakdown makes the problem clear: even with a fair algorithm, we could have an allocation rule that leads to uncomfortable or unjust outcomes, while the outcomes remain fair due to post-processing.

How can we achieve both fairness and justice? Both the individual fairness framework from [3] and our definition of algorithmic justice can be expressed as optimization problems. As in the diagram above, we first consider the mapping $f : \mathcal{X} \to [0, 1]$ from individuals to scores (where we apply the IF framework along with the constraint on the upper bound of the standard deviation σ given by the AJ framework), and then the mapping $g : [0, 1] \to \mathcal{O}$ from scores to outcomes. Thus, the composition $F = g \circ f$ yields the desired mapping from individuals to outcomes.

From the IF framework, we want f to minimize expected loss subject to the Lipschitz condition for (D_∞, d), where $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the given metric. Let $L : \mathcal{X} \times [0, 1] \to \mathbb{R}$ be an arbitrary loss function. As in [3], we denote by $\mathcal{I}$ an instance of our problem consisting of a metric d, a loss function L, and a social welfare function $\mathcal{W}$. Moreover, we write the mapping $\mathcal{M}_f : \mathcal{X} \to \Delta([0, 1])$ as $\mathcal{M}_f = \{\mu_x\}_{x \in \mathcal{X}}$, where $\mu_x = \mathcal{M}_f(x) \in \Delta[0, 1]$.

$$
\begin{aligned}
\text{opt}(\mathcal{I}) := \max_{C \in \mathcal{C}} \Big(\mathcal{W}(S, C) \Big(\min_{\{\mu_x\}_{x \in \mathcal{X}}} \mathbb{E}_{x \sim \mathcal{X}} \mathbb{E}_{a \sim \mu_x} L(x, a) \Big) \Big) \\
\text{where} \quad S = (f(x_1), \ldots, f(x_n)) \\
\text{subject to} \quad \forall x, y \in \mathcal{X} : \quad D_\infty(\mu_x, \mu_y) \leq d(x, y) \\
\forall x \in \mathcal{X} : \quad \mu_x \in \Delta([0, 1]) \\
\forall x \in \mathcal{X} : \quad \sigma(\mathcal{M}_f(x)) \leq \alpha.
\end{aligned}
$$

Figure 2: The Fairness-Justice optimization program.

The Fairness-Justice optimization program is presented in Fig. 2. Then, following the diagram shown in Fig. 1, a fair and just algorithm can be obtained as shown in Fig. 2. As we show in Section 6, we remark that the optimization problem might not have a solution. In that case, we will need to seek trade-offs between fairness and justice.

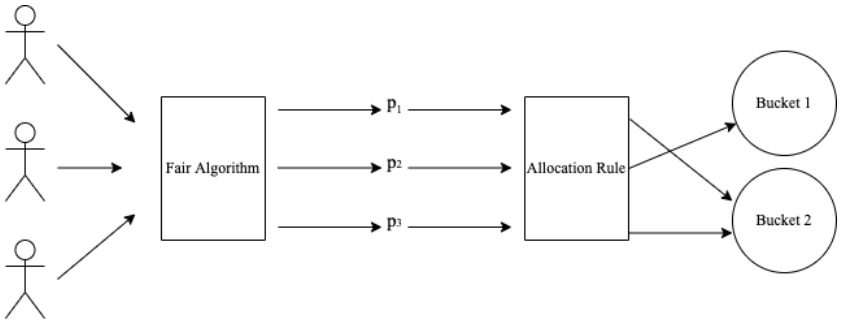

Figure 3: An algorithm that is fair and just (note that the allocation rule is derived directly from the social welfare function).

4. Social Welfare Functions

The notion of justice that we provide in Section 3 involves the use of a social welfare function. Social welfare functions have been well studied [7, 8, 9] and are widely used in welfare economics and policy decision-making. A social welfare function $\mathcal{W}$ is a function over the utility profile U containing the utilities of all concerned parties on a given assignment profile A. Behavioral and welfare economics have historically used social welfare functions (SWFs) as a way to measure the relative desirability of a given distribution of benefits and harms [10].

The framework of individual fairness (IF) [3] and subsequent results related to algorithmic fairness, including outcome indistinguishability (OI) [11] and multicalibration [12], allow algorithm creators to obtain scores that are close to some ground truth (e.g., an individual who will graduate from college with probability 0.95 will receive a score of 0.95 in expectation) and ensure similar treatment of similar individuals. In those settings, the motivation is to model the world as it is while avoiding biases across groups. However, this does not guarantee that the actual decisions or outcomes associated with these predictions are desirable.

For example, suppose we have a class of kindergarteners and, for each student, we can decide whether or not to give the student an ice cream sandwich. Consider two scenarios, one in which we give all the kindergarteners ice cream, and one in which we give none of them the ice cream. A function returning each of these allocations would satisfy the notion of IF (all individuals are

treated identically, which satisfies the requirement that similar individuals are treated similarly) and, because there is no ground truth for whether an individual should receive ice cream, this is not solved by OI or multicalibration. However, these two allocations clearly lead to disparate consequences for the parties involved. Thus, just as in their historical use, we will use SWFs to model the interests of decision makers and agents in allocating limited resources and in determining appropriate assignments to outcomes. In doing so, SWFs can help tell us what to do to move toward a more equitable world.

4.1. Defining Social Welfare Functions

We now provide several formal definitions related to SWFs for a set $\mathcal{X}$ of n individuals.

Definition 4.1 (Score Profile)**.** Given a set of individuals $\mathcal{X}$ such that $|\mathcal{X}| = n$, a *score profile* is the n-tuple

$$S = (s_1, \ldots, s_n),$$

where s_i is the score assigned to individual $x_i \in \mathcal{X}$ by a score function $s : \mathcal{X} \to \Delta[0, 1]$.

Definition 4.2 (Assignment Profile)**.** Given a set of individuals $\mathcal{X}$ such that $|\mathcal{X}| = n$, an *assignment profile* is the n-tuple

$$A = (a_1, \ldots, a_n),$$

where $a_i \in \mathcal{O}$ is the outcome assigned to individual $x_i \in \mathcal{X}$.

Note that the set of possible allocation profiles is not necessarily equal to $\mathcal{O}^n$, since outcomes can be resource-constrained. We write $\mathcal{C} \subseteq \mathcal{O}^n$ to indicate the feasible subset of the outcome space, so we have $A \in \mathcal{C}$.

Definition 4.3 (Utility Function)**.** For a given individual $x_i \in \mathcal{X}$, we define the *utility function* for individual x_i as some function $u_i : [0, 1]^n \times \mathcal{O}^n \to \mathbb{R}$, where, for assignment profiles $A \neq A'$ and score profile S, $u_i(S, A) > u_i(S, A')$ indicates that individual x_i prefers assignment profile A to A'. We define $U = (u_1, \ldots, u_n)$ as the *utility profile* over the set $\mathcal{X}$ of individuals.

Using this definition of a utility function, we can now formally define a social welfare function.

Definition 4.4 (Social Welfare Function)**.** Given a utility profile $U = (u_1, \ldots, u_n)$ over the set $\mathcal{X}$ of individuals, we define the social welfare function $\mathcal{W} : \mathbb{R}^n \to \mathbb{R}$ as a mapping from a vector of individual utilities for a pair $(A \in \mathcal{C}, S \in [0, 1]^n)$ to a real number. We say that, for a score profile $S \in [0, 1]^n$ and assignment profiles $A, A' \in \mathcal{C}$, assignment profile A is socially more desirable than A' if

$$\mathcal{W}(u_1(S, A), \ldots, u_n(S, A)) > \mathcal{W}(u_1(S, A'), \ldots, u_n(S, A')).$$

We often elide U and treat $\mathcal{W}$ as a function $\mathcal{W} : [0, 1]^n \times \mathcal{C} \to \mathbb{R}$ from a pair of an assignment profile and score profile to a real number, essentially incorporating the utility profile U directly into $\mathcal{W}$.

Intuitively, Definition 4.4 provides a framework for finding the comparatively most desirable assignment profile given a set of constraints. In the context of the ice cream example, the above definition of a social welfare function, and the observation that both allocations A of ice cream for all and A' of ice cream for none are individually fair, we can choose the allocation that maximizes the social welfare to find the outcome that is both fair and societally preferred. We define a *choice rule* that models the process of assigning individuals to outcomes based on the scores output by the scoring function.

Definition 4.5 (Allocation Rule)**.** Given a score profile $S \in [0,1]^n$, an *allocation rule* $r : [0,1] \to \mathcal{O}$ maps a score s_i to an assigned outcome $a_i = r(s_i, i)$.

We say that an allocation rule is *social welfare maximizing* if for all individuals $x_i \in \mathcal{X}$ and all possible allocation profiles $A \in \mathcal{C}$ assigned to the individuals, we have:

$$\mathcal{W}(u_1(S, r(s_1, 1)), \ldots, u_n(S, r(s_n, n))) \geq \mathcal{W}(u_1(S, A), \ldots, u_n(S, A)).$$

Item 2 of Definition 3.2 requires that a just algorithm map scores to outcomes according to a social welfare maximizing allocation rule.

Optimization methods such as LP solvers can be used to find the assignment profile $A \in \mathcal{C}$ that maximizes a social welfare function $\mathcal{W}$, and this optimization can be done in the context of constraints imposed by the social planner, the mechanism's design constraints, or limited availability of potential outcomes. Moreover, we can often make complex social welfare functions easier to solve by adding constraints to the optimization problem [9].

4.2. Choosing an Appropriate SWF, and Concrete Examples

We now provide concrete examples of SWFs to illustrate how the use of SWFs can align with notions of justice. Different SWFs offer different notions of justice (and some SWFs may not offer any notion of justice – this is discussed further in Section 8).

The main question surrounding SWFs concerns what should be maximized. Let U be the utility profile. Informally, a SWF $\mathcal{W}(S, A) = \max_{i \in [n]} u_i(S, A)$ only cares about the happiness of the most well-off individual without any concern for the least well-off individual, while an SWF $\mathcal{W}'(S, A) = \min_{i \in [n]} u_i(S, A)$ only cares about the happiness of the least well-off individual while allowing the case in which nobody else is happier than the least-happy individual.

One method for determining the correct SWF is to apply the philosophiocal framework of Rawls' *veil of ignorance* [4]. In this method, an individual performs a thought experiment in which they are presented with a society and asked to come up with a system for mapping individuals in this society to outcomes. However, the individual does not know where in this society they will be positioned when the veil of ignorance is lifted. The idea, then, is that the individual will devise a mapping in which even the worst-off individual is mapped to a reasonable outcome. The resulting mapping will offer a Rawlsian notion of justice in which no punishment is too harsh and no reward too great [4]. A similar idea in the context of algorithmic fairness is described in [13].

Definition 4.6 (Rawlsian Social Welfare Function [8])**.** In *A Theory of Justice* [4], John Rawls argues against the utilitarian perspective and instead provides an alternative based on moral theory that addresses this very issue. Namely, Rawls argues for what he calls the *Difference Principle*, which states that inequalities are unjustified unless they make the least advantaged better off. This leads to the *Rawlsian social welfare function*,

$$\mathcal{W}(S, A) = \min_{i \in [n]} u_i(S, A).$$

Definition 4.7 (Utilitarian Social Welfare Function [14])**.** Jeremy Bentham, often cited as the founder of utilitarianism, was of the opinion that all individuals should be treated the same, regardless of their initial level of utility. The *utilitarian social welfare function*, then, is

$$\mathcal{W}(S, A) = \sum_{i \in [n]} u_i(S, A).$$

More complex relationships and carefully devised metrics can also be modeled by SWFs, including inequality indices like the Gini coefficient.

Definition 4.8 (Gini Coefficient Social Welfare Function [9])**.** The Gini coefficient, which is proportional to the area between the Lorenz curve and a diagonal line representing perfect equality, is a common measure of income inequality in economics [15]. Minimizing inequality of utilities corresponds to maximizing the *Gini coefficient social welfare function*

$$\mathcal{W}(S, A) = 1 - \frac{1}{2\overline{u}(S, A)n^2} \sum_{i,j} |u_i(S, A) - u_j(S, A)|,$$

where $\overline{u}(S, A)$ is the average utility $\overline{u}(S, A) = \frac{1}{n} \sum_{i \in [n]} u_i(S, A)$.

While all of these notions – Rawlsian, utilitarian, and inequality-metric based – have distinct advantages and disadvantages, the fact that they can be formulated as a SWF demonstrates the flexibility of such a model. Namely, the SWFs above offer only a few examples of what a social welfare function can capture. More broadly, we observe that our algorithmic justice framework allows us to strive for equal treatment of equals while also taking into consideration societal concerns and priorities. In this regard, SWF and individual utilities can capture that which many notions of algorithmic fairness, such as multicalibration, cannot: namely, a SWF can offer a model for how we can improve society with respect to a given goal. One natural example of the importance of combining equal treatment of equals with broader societal goals is that of *fair affirmative action* (as considered in, e.g., [16]), which we can also model in our framework as a SWF. In the simplest scenario, including welfare maximization in our definition ensures that when choosing between fair outcomes, we choose the one that is the most preferred by the relevant parties.

Hertweck et al. [17] study other philosophically-inspired notions of SWFs, which they call *patterns of justice*, such as *prioritarianism* and *sufficientarianism*. They also argue for the need to incorporate the notion of utility in algorithmic decision-making, and their work provides an excellent treatment of the philosophical considerations behind the choice of the utility function. Moreover, their definitions focus on group fairness notions, whereas this work is centered around individual fairness. Our work also catalogs a variety of examples that emphasize the shortcomings of fairness definitions which omit a consideration of social welfare. Hence, we believe that the composition of [17] and this work jointly provide a nuanced framework on algorithmic fairness considerations.

4.3. Using SWFs to Assign Outcomes

For illustration purposes, we now present an example of using a SWF and allocation rule to assign individuals to outcomes.

Example 4.9 (Utilitarian Allocation; inspired by [18, pp. 32 and 44])**.** Consider a society with a limited supply of life-saving medication, and suppose the society decides on a utilitarian approach and wants to maximize the number of years added to its citizens' collective life expectancy.

Let the i^{th} coordinate of output from the scoring function capture how many years will be added to life of individual i if they receive the medicine; the social welfare function $\mathcal{W} : [0, 1]^n \times \mathcal{C} \to \mathbb{R}$, then, is chosen according to Definition 4.7 as the sum over $i \in [n]$ of the number of years added to the life of individual x_i under assignment profile $A \in \mathcal{C}$.

The society then finds an assignment profile $A = (a_1, \ldots, a_n) \in \mathcal{C}$ such that, for all $A' \neq A$, $\mathcal{W}(S, A) > \mathcal{W}(S, A')$, and defines the social welfare maximizing allocation rule $r : s_i \times i \mapsto a_i$ such that, for all $i \in [n]$, $r(s_i, i) = a_i$. Assigning medicine according to this allocation rule r, then, results in an assignment profile $A \in \mathcal{C}$ that maximizes the social welfare function $\mathcal{W}$ and therefore maximizes the number of years added to the overall life of the population.

5. What Algorithmic Justice Is Not

In our investigation of injustice in fair and unfair algorithms, we considered whether existing solutions already offer the necessary constraints on algorithms to impose justice. We concluded that, though existing ideas address some issues related to justice, they also lack features necessary for working toward a definition of justice. In particular, justice is not imposed by any of the following.

Different Treatment of Differents. Consider the following example. Suppose a prison sentencing algorithm only has two possible outcomes: "life in prison" and "no prison". Society 1 believes that nobody should be sentenced to life in prison. Assigning everyone to "no prison" optimizes Society 1's social welfare function, even though this means different people are not treated differently (despite having the opportunity to treat different people differently). We feel that different treatment of differents is no more a part of justice than it is a part of fairness: many fair algorithms result in different treatment of differents, but this feature is not a requirement for fairness; likewise, many just algorithms result in different treatment of differents, but this feature is not a requirement.

Accuracy. Justice is not solved by imposing accuracy requirements on an outcome. This is because there is not necessarily a ground truth. Consider an allocation problem in which a society can distribute no more than 1000 pianos to > 1000 individuals. Suppose that anyone with a (fairly computed) "loves pianos" score ≥ 0.5 is eligible to receive a piano. (Note that the use of a threshold does not violate fairness because the scores are determined fairly, and fairness is robust to post-processing.) In a piano-loving society where everyone receives a score ≥ 0.8, there will be some individuals who are eligible for pianos but do not receive pianos, since the size of the population is larger than the number of pianos. In a primarily piano-hating society where one person has a score of 0.8 and everyone else has scores < 0.5, the individual with a score of 0.8 will receive a piano. Because a score of 0.8 does not necessarily correspond to receiving a piano, there is no deterministic ground truth for which scores should result in receiving pianos.

Fair Rankings. Fair rankings deal with ordering a set of candidates based on some predictor. Although fair rankings can be used to solve allocation problems [11], determining whether an algorithm offers justice adds a level *expressiveness* that is lost when simply dealing with rankings. In the piano distribution example, the #1 piano lover in a completely piano-hating society should not be given a piano since they would rather have nothing at all. Rankings alone cannot allow us to determine whether we are in a piano-loving or piano-hating society, which is why the concept of scores is necessary.

Envy-Freeness. Envy-freeness is one of the pillars of fairness in the field of fair division. An agent A envies an agent B if A would prefer to receive B's allocation than to receive their own allocation [19]. An envy-free allocation of resources can be considered fair, since there is no individual who would prefer trading items with another individual over maintaining their current assignment. Importantly, agent A can only be envious of goods that it could have received – if B gets an item that is outside of A's feasible allocation, A doesn't have a value for that item and is thus not envious. A's envy is based on how much A *would have valued* B's bundle and is not necessarily based on how happy B is with their bundle. Envy-freeness, however, is incompatible with individual fairness.

Take again the example of assigning people who have committed crimes to years in prison. Clearly, anyone who receives more prison time would be envious of someone who received less prison time. Thus, we would need to restrict the feasible allocation of years in prison based on the severity of the crime. For example, if we take the case of a shoplifter and a murderer, maybe the murderer can

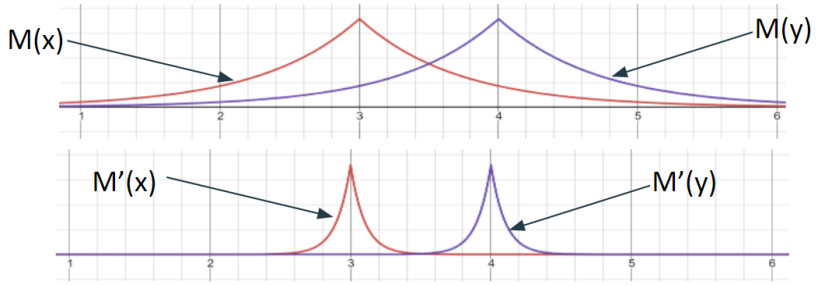

Figure 4: Visual depiction of the intuition behind Theorem 6.2; note that, though the means of distributions are identical, their standard deviations are quite different.

only feasibly be allocated between 50 and 100 years whereas the shoplifter gets allocated at most 1 year. Then, the murderer will not envy the shoplifter, because 1 year is not a feasible allocation for the murderer. The issue is that this makes envy-freeness incompatible with individual-fairness. This is due to the fact that, for individual fairness, if there is an outcome that can be assigned with nonzero probability to at least one individual, then assignment to that outcome must occur with nonzero probability for every individual. If we were only interested in envy-freeness in expectation (i.e. that the expected value of the outcome for an individual is envy free), then issues related to an overly large standard deviation could still arise, where despite being envy-free in expectation, ex-post similar individuals could end up with wildly different treatments [20].

6. The Relationship Between Justice and Fairness

The development of a notion of justice was motivated by the observation that some fair algorithms behave in unjust manners, as presented in Section 2. Because our initial notion of justice was motivated by an exploration of fair algorithms, we now present a comparison of fairness and justice.

6.1. Theorems Relating Fairness and Justice

We now present several theorems that demonstrate how fairness and justice can align, and where these notions differ. Note that Theorems 6.1, 6.2, and 6.3 offer impossibility results that relate justice and fairness.

Theorem 6.1. *Not all fair algorithms are just.*

Proof. We show two ways in which fair algorithms can be unjust.

1. **(Mismatches in standard deviation.)** Suppose f is (D, d)-fair and $\exists x \in \mathcal{X}$ such that $\mathcal{M}_f$ has standard deviation $\geq b$. Then, for standard deviations $a < b$, $\mathcal{M}$ cannot offer a-justice by definition.

   As a concrete example, consider the pair of metrics $d = \varepsilon|x \triangle y|$ (where $|x \triangle y|$ denotes the cardinality of the symmetric distance between x and y), and D_∞.

   Consider an algorithm f that is (D_∞, d)-fair and assigns individuals to scores drawn from the Laplace distribution with scale parameter $b = 1/\varepsilon$, so for an individual x, the score for that individual is $s(x) = S$, where $S \sim Lap(1/\varepsilon)$. We know that S has variance $(1/\varepsilon)^2$.

   Therefore, by Definition 3.2, we cannot have α-justice for $\alpha < 1/\varepsilon$.

2. **(Failure to maximize utility.)** Consider Example 2.3, where a society is seeking to maximize the expected success of college admittees. The admission algorithm described in Example 2.3 is inherently not just with respect to this social welfare function since this admission algorithm does not maximize the social welfare function in expectation.

□

Theorem 6.2. *Not all just algorithms are fair.*

Proof. Suppose j offers 0-justice and has the property that there are $x \neq y \in \mathcal{X}$ such that $j(x) \neq j(y)$ and $d(x, y) < \infty$. Then, by the definitions of D_∞ and individual fairness, j is not individually fair.

More generally, if b is the smallest s.d. such that there exists a (fair) function f for which $D_\infty(\mathcal{M}_f(x), \mathcal{M}_f(y)) \leq d(x, y)$, then for all $a < b$, there is no a-just algorithm which is (D_∞, d)-fair. □

We next show that there is a class of algorithms – namely, 0-just algorithms – that cannot be fair if they map to a nontrivial outcome space $\mathcal{O}$, where we say that $\mathcal{O}$ is nontrivial if $|\mathcal{O}| > 1$ (and where, if $\ell \in \mathcal{O}$ cannot be mapped to by f, we call $\mathcal{O}' = \mathcal{O} \setminus \{\ell\}$ the outcome space, in which case we say that the outcome space is only nontrivial for $|\mathcal{O}'| > 1$).

Theorem 6.3. *No 0-just algorithm mapping to a non-trivial outcome space $\mathcal{O}$ (where non-triviality means that $|\mathcal{O}| > 1$) is fair for (D_∞, d) where $\exists x \neq y \in \mathcal{X}$ such that $f(x) \neq f(y)$ and $0 \neq d(x, y) < \infty$. Note that, if an element $\ell \in \mathcal{O}$ of the outcome space cannot be mapped to by f, we call $\mathcal{O}' = \mathcal{O} \setminus \{\ell\}$ the outcome space.*

Proof. Consider a 0-just algorithm f. Because $\mathcal{M}_f$ has standard deviation 0, f must be deterministic and, because $|\mathcal{O}| > 1$, f must be non-constant. This means there must be two elements $x \neq y$ such that $f(x) \neq f(y)$. By the definition of D_∞, $D_\infty(f(x), f(y)) = \infty$. By assumption, $0 \neq d(x, y) < \infty$, so the inequality $D_\infty(f(x), f(y)) \leq d(x, y)$ cannot hold, so f cannot be fair. □

6.2. Identifying and Correcting Injustice

We now show how our definition of justice in Definition 3.2 can be used to identify unjust algorithms and provide a set of conditions that an algorithm should meet to offer justice.

Example 6.4 (Correcting Example 2.3)**.** Recall Example 2.3, in which there is a fair scoring function $f : \mathcal{X} \to [0, 1]$ with the property that a higher score indicates a higher probability of success in university. For simplicity, let the standard deviation of $\mathcal{M}_f$ for all $x \in \mathcal{X}$ be $\sigma(\mathcal{M}_f(x)) \leq \alpha$.

Suppose the community decides that the "appropriate distribution of the benefits and burdens of social cooperation" is the distribution that offers the most utility to the community as a whole (that is, the community takes a utilitarian view of justice). The community, then, decides that they want α-justice with respect to some utilitarian social welfare function $\mathcal{W}$ defined similarly to the function presented in Definition 4.7, where all individuals $x_i \in \mathcal{X}$ have the same utility function, which is the probability of success in college for individuals who are admitted to college, and is otherwise 0.

We can readily see that the algorithm described in Example 2.3 (which admits the individuals with the lowest probability of success to university) does not maximize the utilitarian social welfare function. In fact, the algorithm described in Example 2.3 minimizes the utilitarian social welfare function, for a given score profile S.

On the other hand, fulfilling Definition 3.2 with respect to $\mathcal{W}$ will ensure that the social welfare function is maximized for a given score profile S, which means that the m individuals who receive the highest success probability scores will be admitted. This resulting assignment profile $A \in \mathcal{C}$ fully aligns with the community's utilitarian approach to justice.

7. Applying Our Definition to Other Notions of Fairness

Section 2 describes algorithms that satisfy individual fairness but behave in a seemingly unjust way. However, since the first paper on individual fairness by Dwork et al. [3], many other notions of fairness have been proposed. In this section, we survey some of these notions to demonstrate that algorithms which satisfy other notions of fairness can also exhibit unjust behavior, and we describe how our definition can prevent these types of behaviors. This shows that our framework is able to address unjust situations in a broad setting and that our results are not restricted to individual fairness notions.

Group Notions. Examples of group fairrness notions include *equalized odds* and *statistical parity* [21]. These notions suffer from the ability to satisfy the constraints with randomized, high-variance algorithms. Additionally, some of these notions admit algorithms which assign individuals to unreasonable outcomes (see Section 2 for examples of injustice that result from the assignment of individuals to unreasonable outcomes): if a model is trained on a dataset with outputs that are the opposite of the desired outputs, that model can satisfy equalized odds while still producing unjust assignments to outcomes.

Multicalibration. Multicalibration was introduced by Hébert-Johnson et al. in 2018 [12] and has since become increasingly popular. In broad terms, multicalibration ensures that, for a defined collection of protected, possibly non-disjoint subsets of the population, an algorithm is calibrated for each of these subsets. However, unjust outcomes, like the unjust taxation example (Example 2.4), can occur when after an algorithm is multicalibrated. Consider a model that is trained on current taxation rates and is then multicalibrated. Similar people are treated similarly by this algorithm, but its behavior is still unjust. Because individuals at the highest incomes pay a lower relative tax rate than the average family, this trend will still be exhibited in the outputs for recommended tax rates that are produced by the multicalibrated algorithm. Similar people are treated similarly by this algorithm, but its behavior is still unjust.

7.1. Generalizing Our Impossibility Results

Versions of the impossibility results described in Section 6 also apply for some of these alternative notions of fairness. Consider a society seeking a major change in the status quo (e.g., in its taxation system). Then, an accurate, multicalibrated algorithm will not be able to be both accurate and just.

More broadly, we observe that these notions do not consider the injustices and biases embedded in the data used in the training nor can be used to improve the status quo (from which the data is gathered). Our proposal of composing fairness with social welfare functions is able to correct for existing injustices.

8. Discussion

In this paper, we have provided several examples demonstrating the limitations of the existing individual fairness framework, as well as the power of our proposed notion of algorithmic justice. This power has been shown in various ways. First, we are able to resolve the issues brought up in our motivating examples. Second, we are able to provide a mathematical definition of our framework and describe it as an optimization program. Third, we envision algorithmic justice as complementary to the IF framework [3], and we show that we can achieve justice and fairness simultaneously while also showing impossibility results concerning the combination of algorithmic justice and fairness.

Lastly, our definition allows us to mathematically prove several desirable properties of algorithmic justice that we were aiming to satisfy.

Although Definition 3.2 offers a useful framework for justice, it has some limitations which we discuss below. We describe situations in which malicious users could exploit the definition to claim "α-justice with respect to social welfare function W" while still offering bad outcomes.

1. Whenever an algorithm is claimed to offer justice with respect to some welfare function, the properties of the welfare function should be evaluated carefully. A social welfare function could be "bad". For example, consider a function that achieves maximum utility when as many journalists as possible are jailed. Although the resulting algorithm is "just with respect to the function that prioritizes jailing journalists", it is clearly unjust with respect to other social welfare functions (such as one that says journalists should never be jailed). One option is to again pull from long-established philosophical notions and use SWFs which have been shown to be appropriate in different contexts.

2. A just algorithm that maps its scores to a small segment of the real number line can offer α-justice for small α while still offering completely arbitrary assignment to outcomes. Auditors, then, should carefully evaluate the context in which the variance is small (or large).

We also remark that our use of SWFs does not imply that establishing SWFs for each particular situation is an easy task (and we are not trying to establish a general SWF as a panacea); rather, we argue for the need in future work to take welfare and utility into account when designing algorithms for societal applications.

Acknowledgements

We would like to thank Cynthia Dwork and Pranay Tankala for their feedback in the initial stages of this research. We would also like to thank our anonymous reviewers for their suggestions.

References

[1] X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, X. Niu, Study on a prediction of P2P network loan default based on the machine learning lightgbm and xgboost algorithms according to different high dimensional data cleaning, Electron. Commer. Res. Appl. 31 (2018) 24–39. URL: https://doi.org/10.1016/j.elerap.2018.08.002. doi:10.1016/j.elerap.2018.08.002.

[2] C. Rudin, C. Wang, B. Coker, The age of secrecy and unfairness in recidivism prediction, CoRR abs/1811.00731 (2018). URL: http://arxiv.org/abs/1811.00731. arXiv:1811.00731.

[3] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, Fairness through awareness, in: Proceedings of the 3rd innovations in theoretical computer science conference, 2012, pp. 214–226.

[4] J. Rawls, A theory of justice, rev. ed. ed., Belknap Press of Harvard University Press, Cambridge, Mass., 1999.

[5] H. Igersheim, A short history of the bergson–samuelson social welfare function, in: Paul Samuelson, Springer, 2019, pp. 279–305.

[6] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, in: Theory of cryptography conference, Springer, 2006, pp. 265–284.

[7] E. Karni, Social welfare functions and fairness, Social Choice and Welfare 13 (1996) 487–496.

[8] L. Hu, Y. Chen, Fair classification and social welfare, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 2020, pp. 535–545.

[9] V. X. Chen, J. Hooker, Welfare-based fairness through optimization, arXiv preprint arXiv:2102.00311 (2021).

[10] J. Finocchiaro, R. Maio, F. Monachou, G. K. Patro, M. Raghavan, A.-A. Stoica, S. Tsirtsis, Bridging machine learning and mechanism design towards algorithmic fairness, in: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021, pp. 489–503.

[11] C. Dwork, M. P. Kim, O. Reingold, G. N. Rothblum, G. Yona, Learning from outcomes: Evidence-based rankings, in: D. Zuckerman (Ed.), 60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019, IEEE Computer Society, 2019, pp. 106–125. URL: https://doi.org/10.1109/FOCS.2019.00016. doi:10.1109/FOCS.2019.00016.

[12] Ú. Hébert-Johnson, M. P. Kim, O. Reingold, G. N. Rothblum, Multicalibration: Calibration for the (computationally-identifiable) masses, in: J. G. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 1944–1953. URL: http://proceedings.mlr.press/v80/hebert-johnson18a.html.

[13] H. Heidari, C. Ferrari, K. P. Gummadi, A. Krause, Fairness behind a veil of ignorance: A welfare analysis for automated decision making, CoRR abs/1806.04959 (2018). URL: http://arxiv.org/abs/1806.04959. arXiv:1806.04959.

[14] O. Stark, M. Jakubek, F. Falniowski, Reconciling the rawlsian and the utilitarian approaches to the maximization of social welfare, Economics Letters 122 (2014) 439–444.

[15] C. Gini, Variabilità e mutabilità, Reprinted in Memorie di metodologica statistica (Ed. Pizetti E (1912).

[16] H. Heidari, J. Kleinberg, Allocating opportunities in a dynamic model of intergenerational mobility, in: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021, pp. 15–25.

[17] C. Hertweck, J. Baumann, M. Loi, E. Viganò, C. Heitz, A justice-based framework for the analysis of algorithmic fairness-utility trade-offs, arXiv preprint arXiv:2206.02891 (2022).

[18] H. Moulin, Fair Division and Collective Welfare, The MIT Press, 2003. URL: https://doi.org/10.7551/mitpress/2954.001.0001. doi:10.7551/mitpress/2954.001.0001.

[19] M.-F. F. Balcan, T. Dick, R. Noothigattu, A. D. Procaccia, Envy-free classification, Advances in Neural Information Processing Systems 32 (2019).

[20] R. Freeman, N. Shah, R. Vaish, Best of both worlds: Ex-ante and ex-post fairness in resource allocation, in: Proceedings of the 21st ACM Conference on Economics and Computation, EC '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 21–22. URL: https://doi.org/10.1145/3391403.3399537. doi:10.1145/3391403.3399537.

[21] A. Chouldechova, Fair prediction with disparate impact: A study of bias in recidivism prediction instruments, Big data 5 (2017) 153–163.