Fairness and Abstraction in Sociotechnical Systems

Andrew D. Selbst
Data & Society Research Institute
New York, NY
andrew@datasociety.net

danah boyd
Microsoft Research and
Data & Society Research Institute
New York, NY
danah@datasociety.net

Sorelle A. Friedler
Haverford College
Haverford, PA
sorelle@cs.haverford.edu

Suresh Venkatasubramanian
University of Utah
Salt Lake City, UT
suresh@cs.utah.edu

Janet Vertesi
Princeton University
Princeton, NJ
jvertesi@princeton.edu

ABSTRACT

A key goal of the fair-ML community is to develop machine-learning based systems that, once introduced into a social context, can achieve social and legal outcomes such as fairness, justice, and due process. Bedrock concepts in computer science—such as abstraction and modular design—are used to define notions of fairness and discrimination, to produce fairness-aware learning algorithms, and to intervene at different stages of a decision-making pipeline to produce "fair" outcomes. In this paper, however, we contend that these concepts render technical interventions ineffective, inaccurate, and sometimes dangerously misguided when they enter the societal context that surrounds decision-making systems. We outline this mismatch with five "traps" that fair-ML work can fall into even as it attempts to be more context-aware in comparison to traditional data science. We draw on studies of sociotechnical systems in Science and Technology Studies to explain why such traps occur and how to avoid them. Finally, we suggest ways in which technical designers can mitigate the traps through a refocusing of design in terms of process rather than solutions, and by drawing abstraction boundaries to include social actors rather than purely technical ones.

CCS CONCEPTS

• **Applied computing** → **Law, social and behavioral sciences**;
• **Computing methodologies** → *Machine learning*;

KEYWORDS

Fairness-aware Machine Learning, Sociotechnical Systems, Interdisciplinary

ACM Reference Format:
Andrew D. Selbst, danah boyd, Sorelle A. Friedler, Suresh Venkatasubramanian, and Janet Vertesi. 2019. Fairness and Abstraction in Sociotechnical Systems. In *FAT* '19: Conference on Fairness, Accountability, and Transparency (FAT* '19), January 29–31, 2019, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3287560.3287598

FAT '19, January 29–31, 2019, Atlanta, GA, USA*

ACM ISBN 978-1-4503-6125-5/19/01…$15.00
https://doi.org/10.1145/3287560.3287598

1 INTRODUCTION

On the typical first day of an introductory computer science course, the notion of abstraction is explained. Students learn that systems can be described as black boxes, defined precisely by their inputs, outputs, and the relationship between them. Desirable properties of a system can then be described in terms of inputs and outputs alone: the internals of the system and the provenance of the inputs and outputs have been abstracted away.

Machine learning systems are designed and built to achieve specific goals and performance metrics (e.g., AUC, precision, recall). Thus far, the field of fairness-aware machine learning (fair-ML) has been focused on trying to engineer fairer and more just machine learning algorithms and models by using fairness itself as a property of the (black box) system. Many papers have been written proposing definitions of fairness, and then based on those, generating best approximations or fairness guarantees based on hard constraints or fairness metrics [24, 32, 39, 40, 72]. Almost all of these papers bound the system of interest narrowly. They consider the machine learning model, the inputs, and the outputs, and abstract away any context that surrounds this system.

We contend that by abstracting away the social context in which these systems will be deployed, fair-ML researchers miss the broader context, including information necessary to create fairer outcomes, or even to understand fairness as a concept. Ultimately, this is because while performance metrics are properties of systems in total, technical systems are subsystems. Fairness and justice are properties of social and legal systems like employment and criminal justice, not properties of the technical tools within. To treat fairness and justice as terms that have meaningful application to technology separate from a social context is therefore to make a category error, or as we posit here, an abstraction error.

In this paper, we identify five failure modes of this abstraction error. We call these the Framing Trap, Portability Trap, Formalism Trap, Ripple Effect Trap, and Solutionism Trap. Each of these traps arises from failing to consider how social context is interlaced with technology in different forms, and thus the remedies also require a deeper understanding of "the social" to resolve problems [1]. After explaining each of these traps and their consequences, we draw on

the concept of *sociotechnical systems* to ground our observations in theory and to provide a path forward.

The field of Science and Technology Studies (STS) describes systems that consist of a combination of technical and social components as "sociotechnical systems." Both humans and machines are necessary in order to make any technology work as intended. By understanding fair-ML systems as sociotechnical systems and drawing on analytical approaches from STS, we can begin to resolve the pitfalls that we identify. Concretely, we analyze the five traps using methodology from STS, and present mitigation strategies that can help avoid the traps, or at the very least help designers understand the limitations of the technology we produce. Key to this resolution is shifting away from a solutions-oriented approach to a process-oriented one—one that draws the boundary of abstraction to include social actors, institutions, and interactions.

This paper should not be read as a critique of individual contributions within the field of fair-ML, most of which are excellent on their own terms. Rather, as scholars who have been invested in doing this work, we take aim at fair-ML's general methodology, limited as it is by the bounds of our primarily technical worldview. We echo themes of STS scholars such as Lucy Suchman [66], Harry Collins [17], Phil Agre [5], and Diana Forsythe [35]—who in the 1990s critiqued the last wave of artificial intelligence research along similar lines, highlighting that the culture of and modes of knowledge production in computer science thwarted the social goals that the field was trying to achieve. An emerging wave of similar critiques has been directed at the field of fair-ML [9, 29, 36]. We hope to offer a constructive reform in our nascent field by highlighting how technical work can be reoriented away from solutions to process and identifying ways in which technical practitioners and researchers can meaningfully engage social contexts in order to more effectively achieve the goals of fair-ML work.

2 THE ABSTRACTION TRAPS

Abstractions are essential to computer science, and in particular machine learning. The ubiquity of machine learning across so many domains comes from the way in which the domain-specific aspects of the problem—broadly, the social context—are abstracted so that machine learning tools can be applied. In this section, we identify and explain five different traps: failure modes that result from failing to properly account for or understand the interactions between technical systems and social worlds.

2.1 The Framing Trap

Failure to model the entire system over which a social criterion, such as fairness, will be enforced

The most common abstractions in machine learning consist of choosing representations (of data), and labeling (of outcomes). Once these choices are made, they constitute the description of what we call the *algorithmic frame*. Within this frame, the efficacy of an algorithm is evaluated as properties of the output as related to the input. For example, does the algorithm provide good accuracy on training data and good generalizability to unseen data from the same distribution? We refer to this as an end-to-end property of the frame; it is how any particular algorithm is evaluated. Yet, while in the algorithmic frame (where the vast majority of current data science resides [38]), the abstraction is taken as given and is rarely if ever interrogated for validity. This is despite the fact that abstraction choices often occur implicitly, as accidents of opportunity and access to data.

Within the algorithmic frame, any notion of "fairness" cannot even be defined. This is because the goal of the algorithmic frame is to produce a model that best captures the relationship between representations and labels. Investigating the idea of fair machine learning requires us to expand the frame to encompass not just the algorithm but the algorithm's inputs and outputs. We refer to this abstraction level as the *data frame*. The significance of this change in abstraction is that what used to be part of the interface—the inputs and outputs—is now brought into the system and can be interrogated directly. Fair-ML algorithms explicitly investigate ways in which the choices of representations and labels might affect the resulting model. The end-to-end-guarantee, in addition to local guarantees provided by the (now-embedded) algorithmic frame, is a formal measure that seeks to approximate a socially desirable goal like fairness.

Prioritizing the data frame is the first sign that there is a larger social context to what appeared to be purely technical in the algorithmic frame. Indeed, much of the literature on fair-ML can be described as making this conceptual shift. Conceptualizing "fairness" requires that we consider other characteristics of the data such as demographic information. For example, the earliest fair-ML work by Ruggieri et al. formulates "fairness" as a goal for model optimization [59]. Others like Feldman et al. [24] argue that fair outcomes can be achieved by removing bias in training data prior to training a model. Such examples postulate a data frame in which the fair-ML community may focus their efforts.

Whereas the data frame opens up consideration of the inputs and outputs of the learned model, it is still an attempt to eliminate larger context and abstract out the problems of bias in a mathematical form. By contrast, a *sociotechnical frame* recognizes explicitly that a machine learning model is part of a sociotechnical system, and that the other components of the system need to be modeled. By moving decisions made by humans and human institutions within the abstraction boundary, fairness of the system can again be analyzed as an end-to-end property of the sociotechnical frame.

Consider the question of labels in a risk assessment setting. Broadly speaking, a risk assessment tool is a predictive model to assess the "risk" of a defendant to aid in decision-making at various stages of the criminal justice pipeline. Risk assessment tools are used at arraignment to determine whether defendants should be released pretrial and whether bail should be required [68]. They are also used at sentencing and parole hearings, but we will focus here on the pretrial setting.

A common goal for a risk assessment is to determine whether a defendant will fail to appear in court for relevant hearings. Occasionally, a secondary goal is to determine whether there is a risk that the defendant will commit other crimes while under pretrial release. But though those stated goals are the outputs of the risk assessment model, they are not outputs of the criminal justice system, and are therefore not truly the question that determines fairness. Ultimately, at an arraignment, a defendant is released on their own recognizance, released on bail, or detained. These are the important

human outcomes by which we measure fairness or justice in the system.

In most jurisdictions, risk assessment scores are presented to a judge as a recommendation. Yet, judges do not consistently take the recommendations into account [15, 65]. Different judges might exhibit automation bias [16, 62], deviate from recommendations in biased ways, or ignore them entirely. Failure to account for how judges respond to scores creates a problem for risk assessment tools that come with fairness guarantees. Such a tool might present a guarantee of the form "if you use these thresholds to determine low, medium and high risks, then your system will not have a racial disparity in treatment of more than X%". But if a judge only adopts the tool's recommendation some of the time, the claimed guarantee might be incorrect, because a "shifted" threshold caused by judicial modification comes with a much poorer effective guarantee of fairness. Moreover, if the judge demonstrates a bias in the types of cases on which she alters the recommendation, there might be no validity to the guarantee at all. In other words, a frame that does not incorporate a model of the judge's decisions cannot provide the end-to-end guarantees that this frame requires.

2.2 The Portability Trap

Failure to understand how repurposing algorithmic solutions designed for one social context may be misleading, inaccurate, or otherwise do harm when applied to a different context

One reason designers might fall into the Framing Trap is because computer science culture prizes and often demands portability. Transferrable code, purposefully designed to be as abstract as possible, is considered more useful (because it is reusable), skillful, elegant, or beautiful. This imperative is ingrained strongly in almost anyone trained as a computer scientist or engineer, and suggests that design will first aim to create tools independent of social context.

Portability is no less important in machine learning than in other software domains. In fact, the structure of machine learning yields a very simple expression of abstract design. Problems are categorized by the nature of the learning task to be solved (e.g., classification, clustering, reinforcement learning, regression). This task-centric abstraction is key: it allows the same "solution" (e.g. a better algorithm for binary classification) to appear to be applicable to problems in a variety of social settings—whether predicting risk of recidivism, loan default, or being a bad employee—regardless of the different social context around these questions. Indeed, the vast codebase of tools for doing machine learning (e.g., scikit-learn, Rstats) and platforms for deep learning (e.g.,tensorflow, pyTorch) are explicitly designed to encourage this portability. The problem "enters" the system as data and exits the system as a prediction.

The fair-ML literature, even as it has moved beyond the algorithmic frame, has still embraced portability as a core value. For example, many of the papers that seek to provide "fair" solutions to machine learning tasks fix a definition of fairness as a portable module, and then seek to optimize a cost function that combines this definition of fairness with standard notions of classifier accuracy [40, 72]. Other papers fix a definition of fairness and then seek to modify training data in order to prevent a black-box classifier (itself a portable module) from making "unfair" decisions [24]. Indeed, a recent well-regarded paper is a stand-out example of abstract and portable design: the entire paper describes a process for building a "fair wrapper" around any classifier to make the resulting outputs fair [4].

Each of these papers make certain assumptions about the world—as they must for the sake of the formalism. For example, Friedler et al. [26] discuss two: that the data observed is close enough to the facts that matter (WYSIWYG) or that groups are, on the whole, similar to each other with respect to the task, and thus "we're all equal." Certain assumptions will hold in some social contexts but not others. The assumptions should reflect the anticipated application.

Suppose, then, that the social context is modeled well enough, by including approximations of the relevant humans and human institutions in the model—avoiding the Framing Trap—and that the axioms that are true for the context are chosen. Then by taking the social context into account, absorbing parts of the court or employment system into the model, and making assumptions specific to the social context—avoiding the Formalism and Ripple Effect Traps as discussed below—the designer has created a system that is not portable between social contexts. Although designers are taught from an early stage that portability is the ultimate goal of system design, with social objectives, this produces the Portability Trap. To design systems properly for fairness, one must work around a programmer's core programming.

There are two additional points to note. First, the problem is not just about shifts in domain (e.g., from automated hiring to risk assessments). Even within a domain, such as between court jurisdictions, the local fairness concerns may be different enough that systems do not transfer well between them. Second, while frameworks like domain adaptation and transfer learning do provide a limited degree of portability between contexts, they encode context merely as shifts in the joint distribution of features and labels. This is not sufficiently expressive to capture the vast changes in social context that might occur between domains.

2.3 The Formalism Trap

Failure to account for the full meaning of social concepts such as fairness, which can be procedural, contextual, and contestable, and cannot be resolved through mathematical formalisms

Perhaps the issue that has received the most attention in the fair-ML literature has been the definition of fairness. Because algorithms "speak math," the mandate within the fair-ML community has been to mathematically define aspects of the fundamentally vague notions of fairness in society in order to incorporate fairness ideals into machine learning. Feldman et al. [24], for example, formalize the Equal Employment Opportunity Commission's (EEOC) 80% rule into a formal measure of bias that they call disparate impact. The well-known debate about the COMPAS risk score pitted journalists at ProPublica who demonstrated that a certain measure of fairness (equality of error rates) was violated [7], against Northpointe, the creators of COMPAS, who argued that it was fair because the test accuracy was equalized across groups [19].

Fully describing the myriad proposed definitions is out of scope of this work. Each of these definitions, however, are simplifications that cannot capture the full range of similar and overlapping notions of fairness and discrimination in philosophical, legal, and sociological contexts. Limiting the question to a mathematical formulation gives rise to two distinct problems in practice.

First, there is no way to arbitrate between irreconcilably conflicting definitions using purely mathematical means [14, 42]. Assuming that any of the mathematical definitions are appropriate, social context must dictate the choice. Consider an automated resumé screen: we might be less concerned with false negatives than false positives because there is more filtering at the back end (the interview itself). Where false negatives end the process entirely, closing out particular candidates, the consequence of false positives is a little extra work for the employer. In the criminal justice context, however, we might be most concerned about equalizing false positives, which result in keeping people locked up and where disparities will further entrench a minority-dominated prisoner underclass. Whether one agrees with these particular normative judgments about the balance between false positives and negatives, it should be clear that the normative values contained within the relevant social context are the determinants.

The second problem with formalization arises from the concern that *no* definition might be a valid way of describing fairness. Fairness and discrimination are complex concepts that philosophers, sociologists, and lawyers have long debated. They are at times procedural, contextual, and politically contestable, and each of those properties is a core part of the concepts themselves.

Procedurality. The biggest difference between law and the fair-ML definitions is that the law is primarily procedural and the fair-ML definitions are primarily outcome-based. If an employer fires someone based on race or gender, it is illegal, but firing the same person is legal otherwise, despite the identical outcome [73]. Disparate impact is similarly procedural. The EEOC's 80% guideline is the first step in a doctrine that then asks the defendant whether a test was "job related ... and consistent with business necessity," then whether the plaintiff can show an equally effective but less discriminatory alternative that the defendant refused to use [67]. Thinking about disparate impact as the 80% threshold alone is incorrect.

Contextuality. Legal scholar and philosopher Deborah Hellman has argued that what we mean by "discrimination" is actually wrongful discrimination: we make distinctions all the time, but only cultural context can determine when the basis for discrimination is morally wrong [34, pp. 28–29]. Sociologist and legal scholar Issa Kohler-Hausmann has similarly argued that discrimination "can only be comprehended with access to situated cultural knowledge about the relevant categories that make up a particular society's system of stratification and a normative critique of how those categories operate." [45, p. 7]. Though law should not ultimately be our yardstick because it regularly fails to adequately incorporate ideas about discrimination that are accepted by sociology [8] or behavioral psychology [46, 51], even existing law relies on the deeply contextual nature of the question, with differences depending on attribute (e.g. race, gender, disability, religion), sphere of society (e.g. school, employment, housing) and discriminator (e.g. public versus private and size, type, or mission of organization) [73].

Contestability. Discrimination and fairness are politically contested and shifting. Legislation and court cases change legal definitions, and advocacy and culture change social norms. The foundations of liberal society depend on the idea that some concepts will be fundamentally contestable and will shift over time, that communities should be allowed to collectively define norms and laws. To set them in stone—or in code [52]—is to pick sides, and to do so without transparent process violates democratic ideals.

That the mathematical definitions eliminate these nuances is a fact known to many fair-ML researchers [14, 24, 31, 53]. But that each of these attributes are core components of the concepts of fairness and discrimination has been underappreciated thus far.

2.4 The Ripple Effect Trap

Failure to understand how the insertion of technology into an existing social system changes the behaviors and embedded values of the pre-existing system

When a technology is inserted into a social context, it has both intended and unintended consequences. Chief among the unintended consequences are the ways in which people and organizations in the system will respond to the intervention. To truly understand whether introduction of the technology improves fairness outcomes, it is not only necessary to understand the localized fairness concerns, as discussed above, but also how the technology interacts with a pre-existing social system.

For example, without risk assessments, probation officers make recommendations to judges about release based on their observations about the defendant, and the judge makes her own determination about likelihood of a defendant's failure to appear. These judgments are inherently riddled with potential for bias (which motivates technical interventions). With risk assessments, however, judges see mathematically calculated scores and recommendations before making a decision. Avoiding the Ripple Effect Trap requires us to account for how the technical system can affect the judge's behavior, the judge's perception of her role and expertise, and the relative power dynamics between the judge and other actors in the context. Judges' responses to risk assessments may be constrained or discretionary, depending on jurisdiction [64]. Judges might defer to the risk assessment because it appears more rigorous, or they might resist the influence of private companies on their domain. How scores are initially used might differ from what happens when judges see them frequently [65]. Political pressure and other social forces might influence judges over time in ways they themselves don't recognize. Understanding how the intervention affects the context also requires evaluating not just the abstract possibility of the technology but what unfolds in situ because the effects of the technology will change based on the particulars of the social context; the responses of actors to technical interventions in the employment and criminal justice contexts will differ.

Technologies can also alter the underlying social values and incentives embedded in the social system. Specifically, new tools offer the possibility of unconsciously privileging quantifiable metrics. Switching from bail to sentencing, the purpose of risk assessments becomes identifying dangerousness so as to prevent releasing someone who is likely to reoffend. But incapacitation is only one of at

least seven existing rationales for punishment: prevention, incapacitation, rehabilitation, deterrence, education, retribution, and restoration [47, § 1.5]. These rationales are always in tension; a potential consequence of risk assessments is privileging the social value that is quantified [21]. More attention is needed to understand when technologies trigger value shifts in social systems.

2.5 The Solutionism Trap

Failure to recognize the possibility that the best solution to a problem may not involve technology

Because fair-ML is rooted in computer science, there is no concept of the system without a technical intervention. This might be recognized as the "law of the instrument" (colloquially described as "if you have a hammer, everything looks like a nail") [41] or "technological solutionism" [56]. In the best case, fair-ML system creators can iteratively work outwards to improve a model, encompassing more and more social context as required. When done well, this process may approximately model enough of the social environment that useful claims about fairness can be made. But by starting from the technology and working outwards, there is never an opportunity to evaluate whether the technology should be built in the first place [11].

There are two broad situations in which starting with technology may be the wrong approach, or rather, one where modeling the situation will not work no matter how many approximations one makes. The first is when fairness definitions are politically contested or shifting, as described in the Formalism Trap. Modeling requires pinning down definitions. Code calcifies. When fairness is a politically contested, movable issue, a model may not be able to capture the facets of how it moves. The second is when the modeling required would be so complex as to be computationally intractable.

To understand whether to build, we must also understand the existing social system. In the risk assessment context, we need to know how arresting officers, prosecutors, and judges introduce bias into both the process and the data. We need to understand how social factors (e.g., poverty) shape criminal activity as well as the contested ideas of the role of criminal justice in the first place. We need to understand how concepts of fairness that surround assessing someone's risk are political, contested, and may shift over time. More concretely, this leads to questions about, for example, whether judges are elected and responsive to political shifts, whether "failure to appear" is culturally and practically a proxy for being poor, or how demographics of the jurisdiction may change in the near future. If the fairness concept is contested or shifting, it might not be easily be modeled.

One might think that the uncertainty could itself be modeled, and that leads to the second issue. When there is not enough information to understand everything that is important to a context, approximations are as likely to make things worse as better. This could occur because some of the aforementioned traps have not been resolved or because there is not enough empirical evidence to know. In such a case—and especially when the stakes are high, as they are in criminal justice—it is prudent to study what might happen before implementing a technology simply based on its potential to improve the situation.

But it could also be that a particular system relies on unmeasurable attributes. Trying to predict how politics will change is difficult. Human preferences are not rational and human psychology is not conclusively measurable [63]. Whether one is trying to model political preference or something else, the system could just be too complex, requiring a computationally and observationally impossible amount of information to model properly. In that case, there should be heavy resistance to implementing a new technology at all.

3 A SOCIOTECHNICAL PERSPECTIVE

Achieving fairness in machine learning systems requires embracing a sociotechnical view. That is, technical actors must shift from seeking a solution to grappling with different frameworks that provide guidance in identifying, articulating, and responding to fundamental tensions, uncertainties, and conflicts inherent in sociotechnical systems. In other words, reality is messy, but strong frameworks can help enable process and order, even if they cannot provide definitive solutions.

The wide range of findings in the sociotechnical systems literature have important implications for the design of ML systems that are meant to interface with social life and produce socially beneficial outcomes. Chief among these insights is that the social must be considered alongside the technical in any design enterprise. For example, it is as important to understand the behaviors of hiring managers and job candidates when using an automated resumé screen as it is to understand the role of the software. The same is true for how judges use risk assessment tools.

But how can this be done effectively? Social systems are large, sprawling, and unruly, while technologies are surely more distinct and tractable. Sociotechnical thinking can allow us to gain purchase on the problem, first by realizing that technologies only appear to be distinct and tractable. In reality, technology always involves social actors [48]. Second, certain theories from STS can be read as encouragements to build differently. They allow us to see the otherwise invisible actors, social groups, and pressures involved, to identify forks in the technological road, and to make design choices responsibly [57]. This is because STS approaches shed light on the misconception that technology advances of its own accord, along some predetermined path or to satisfy optimal efficacy. Instead, different social environments, actors, and social groups shape the kinds of technology that eventually becomes successful. In what follows, we return to the traps previously discussed and in each case explain how an STS perspective points the way to understanding the trap and avoiding it.

3.1 An STS Lens on the Framing Trap

Adopting a "heterogeneous engineering" approach

If the Framing Trap is the result of choosing only certain technical parts of the system to model and manage, a more effective way forward is to consider both human activities and machine ones at the same time [49]. This is what sociologist John Law calls "heterogeneous engineering" [50]. Heterogeneity here refers to the requirement that we think simultaneously about what different technical parts of the apparatus will do, and what the humans that operate, live alongside, and otherwise contribute to them will do

as well. This does not mean that we can engineer human decisions. Rather, heterogeneous engineering suggests that we draw the boundaries of abstraction to include people and social systems as well, such as local incentives and reward structures, institutional environments, decision-making cultures and regulatory systems. Working with multiple elements of the sociotechnical puzzle allows us to model (and hopefully, produce) "fair" outcomes as properties of systems that are sociotechnical through and through.

To see how heterogeneous engineering works, consider the cell phone. While phones appear to be distinct, technological objects, they require a network of many technical and social elements to operate: from satellites, wireless protocols, batteries, chargers, and electrical outlets to companies like Apple, Google, or Verizon, regulatory agencies like the FCC, and standards setting organizations like the IEEE. It is only when all these elements are assembled and working together effectively that the phone turns on, connects to a network, and lets a user make a call. By conceptually separating machine learning from the social context in which it is operating, those invested in fair-ML risk making the same categorical mistake as a company that designs a cell phone without knowledge of data plans, satellites, regulators, or anyone else to call.

Of course, designing for all social exigencies is an overwhelming and impossible task. "Successful large scale heterogeneous engineering," explains Law, "is difficult" precisely because the various pieces of the sociotechnical puzzle always threaten to shift or move away [50, p. 114]. A wireless company upgrades its network; an IEEE meeting sets a new standard; a hurricane knocks out power for three weeks. But this, according to Law, is all the more reason to think sociotechnically. Because such systems are inherently fragile and complex, ignoring certain elements of the network or assuming that they are too unruly or unpredictable to incorporate undermines the ability of the system to operate as intended. As Law puts it, "we must be ready to handle heterogeneity in all its complexity, rather than adding the social as an explanatory afterthought." [50, p. 117]

The heterogeneous engineering perspective on sociotechnical systems combats the Framing Trap by suggesting that we draw our analytical boxes around both human and technical components. Because fairness cannot exist as a purely technical standard, thinking about people and their complex environments—such as relationships, organizations, and social norms—as part of the technical system from the beginning will help to cut down on the problems associated with framing and solutionism. Of course, it would eliminate the benefits of abstraction to include the *entire* network of people and things that interact with the fair-ML tools. But drawing our black box boundary around at least one technical element and one social element—a social institution, an organizational context, a regulatory necessity, a possibility of human action—will give better results while still enabling some tractability on the problem. Equally importantly, recognizing which parts of this sociotechnical system are in focus when evaluating for fairness is crucial for communicating the boundaries of the fairness guarantee.

3.2 An STS Lens on the Portability Trap

Contextualizing user "scripts"

Heterogeneous systems thinking [13, 48, 50] also addresses the Portability Trap. Take the cell phone out of its sociotechnical network—where there is no cellular coverage or where the company does not have a roaming contract with a local carrier, for instance—and it will not work [6]. This was anthropologist Madeleine Akrich's central realization when she studied how light bulbs and generators, developed in France as part of a development project, failed once imported to West Africa. The engineers had designed with different standards for electric ports in mind, but did not consider how power generators might be shared in rural villages or how electricity was metered and paid for. Akrich realized that the user "scripts" that dictate how technologies are supposed to be used only work if all the social and technical elements of a network are assembled properly.

A technology may be designed with many use cases in mind, but in each case, a designer or computer scientist hopes to embed certain "scripts" for action into their product. In our case, the scripts are nothing less than producing fair outcomes in social contexts as varied as hiring employees and assessing someone's risk to society. But the theory of scripts shows that such outcomes will always be disrupted as soon as the code, device, or software moves to a different context. At the very least, the code will be taken up in a new context that shifts the outcome of the system altogether to one that may or may not be fair.

Scripts also demonstrate, for the fair-ML researcher, that concepts such as "fairness" are not tied to specific objects but to specific social contexts. While fair-ML engineers may be tempted to make their code abstract and portable, attaching the label "fair" to the code will erroneously encourage others to appropriate this code without understanding how the script changes or is disrupted with a shift in social context. In other words, the theory of scripts shows how a portable orientation will almost always undo the very possibility of fairness that makes the code desirable.

3.3 An STS Lens on the Formalism Trap

Identifying "interpretive flexibility," "relevant social groups," and "closure"

Avoiding the formalism trap seems to require a rejection of mathematical solutions, but this is not necessarily the case. Instead, we should consider how different definitions of fairness, including mathematical formalisms, solve different groups' problems by addressing different contextual concerns. Here, the Social Construction of Technology program (SCOT) developed by sociologist Trevor Pinch and historian Wiebe Bijker offers relevant ways forward [57].

Social constructivism describes how technology is developed, made sense of, and adopted in social contexts, with human users at the forefront. The key elements of the SCOT framework are a period of *interpretive flexibility* experienced by *relevant social groups*, followed by *stabilization*, and eventually *closure*. As Pinch and Bijker describe, when a new technology is proposed, many different versions are produced, built, and sold. While there is no agreed-upon version of what constitutes success, the value and use cases for each version are open to interpretation. Pinch and Bijker show that different interpretations emerge, each advanced by a *relevant social group*: a group in society that has a specific idea of what problems the technology needs to solve. Designers

respond by producing different versions of the tool that eventually are deemed to solve the local problem, ultimately stabilizing the artifact in question. Which version wins out is a question of whether or not that relevant social group considers the problem solved: this produces *closure*.

SCOT suggests that social groups, including users and designers, have the power to shape technological development. We are currently in a period of interpretive flexibility in the fair-ML community. There is agreement among several social groups that algorithmic bias is a problem that needs redress. Computer science researchers, as one relevant social group, are choosing between different fairness formalizations with distinct solutions. As fair-ML researchers seek to define the "best" approach to fairness, we also implicitly decide which problems and relevant social groups are important to include in this process. Our choices prioritize certain views over others, exerting power in ways that must be accounted for. We may privilege the needs of people in our community—technical practitioners aiming to have precise modules of portable code or technical academics who need to publish innovative algorithms—over those impacted by the use of fair-ML algorithms.

Recognizing that our version of "fairness" is only one interpretation of the problem is critical for considering potential solutions that may address the needs of other relevant social groups. Outside of the research community, some stakeholders—such as companies building risk assessment tools—are also seeking closure that stabilizes their approach to fairness, while others—usually those subject to the effects of technology—have little voice at all [23]. Because the discourse in industry and academia shapes the resulting technologies, and accordingly the standards, it seems likely that fair-ML practitioners and researchers will be influential in achieving closure. At the same time, our community might be rendered accidental kingmakers by privileging the fairness-related concerns of certain social groups with access: groups who become "relevant" simply by means of their existing relationships to fair-ML researchers or by means of an existing voice in society. Designers can and must make informed choices as to who—not just what—they will listen to and include as they seek to produce fair outcomes.

The SCOT literature documents many examples in which the power of a relevant social group impacted closure, ranging from the Model T Ford to the electric refrigerator, DRM, and Betamax [18, 28, 30, 43]. In each case, the technologies that "won" did so not because they were technically superior to their competition, or solved actual users' problems, or even because their uptake was subject to the free market—but because of powerful companies or actors with vested interests in their development. Closure is not always achieved when the best solution is found; it is typically a byproduct of other social mechanisms.

More common is *rhetorical closure*, which occurs when the relevant social groups describe the problem as solved, and move on. In some cases, one design is deemed to achieve this goal, while other functional measures are said to not matter if this goal is achieved (i.e. if the algorithm is fair, does it matter if it is fast?). In other cases, individuals redefine the problem such that the solution they already have at hand, or can easily create, becomes the solution to a problem (i.e. if the algorithm runs the fastest, does it matter if it is only passably fair?). Pinch and Bijker call this *closure by redefinition of the problem*. A danger of the Formalism Trap is that the assumption that "fairness can be defined mathematically" may become a general agreement amongst those creating the technology, leading designers to assume that a system is fair when it has a mathematical model of fairness built into it. This is rhetorical closure because such systems may not produce fair outcomes when placed into contexts of use. It redefines the problem of fairness to one that a computer can solve.

Ultimately, SCOT allows engineers to see that the social world is a mechanism that fundamentally shapes technical development at every level. This recognition allows us to intervene and shape technologies in line with the concerns of different relevant social groups. It also demonstrates that the problems associated with formalism are not simply problems of finding more, better, or different mathematical expressions. They are instead related to underlying assumptions about who can solve the problems of fairness and how, which other problems must be solved, and which social groups are deemed relevant in the process. They are also related to closure mechanisms at play that are not "solving" the problem of fairness at all, but rather re-defining the problem space such that it can be solved. By defining fairness as a problem that can be resolved mathematically, without reference to context and by computer scientists alone, the problem of fairness can look like it is solved cleanly—but only because it has been defined so narrowly, and by a certain social group competing for relevance.

3.4 An STS Lens on the Ripple Effect Trap

Avoiding "reinforcement politics" and "reactivity"

It may seem impossible to predict what will change when a technology enters a social context. Fortunately, several common changes are well documented and understood in the literature. Awareness of these common "ripple effects" in advance can alert the fair-ML practitioner to avoid common pitfalls that may negatively affect the fairness of their proposed systems.

First, countless studies of new technologies emphasize how existing groups use the occasion of this new technology to reinforce or argue for power and position. Computer scientist Rob Kling calls this process *reinforcement politics* [44]. We typically see such reinforcement of political power when management purchases software for monitoring or otherwise controlling subordinate groups in an organization. In other cases, new technologies become opportunities to argue for more power in an organizational context. Sociologist Steve Barley noted that when CT scanners were introduced in two otherwise identical hospitals, the devices became a resource in an existing power struggle between radiologists and technicians over who could manage the machine or interpret its results [10].

The designers of CT scanners or marketing software no doubt did not intend for their technology to reproduce organizational inequalities, any more than fair-ML researchers might expect their risk assessment software to produce arguments between judges, court clerks, and technical experts. Introducing a new technology may appear to alter an organization's dynamics but may in fact aid in reifying a pre-existing group's claim to power, while downplaying or downgrading other groups' authority.

Second, studies of monitoring, evaluation, and measuring technologies demonstrate that they produce *reactivity behaviors*, thus

altering the very social context that the original design was meant to support [15, 22, 25]. This may undermine social goals, exacerbate old problems and actively introduce new ones. Classic examples of this include how individuals introduce and then proceed to game credit scores or publication counts. The COMPAS risk assessment was based in part on a questionnaire that asked about criminal history, drug use, and "criminal personality" [7]. Some of the questions—for example, how many of one's friends have been arrested—seem likely to indicate greater criminality, so it would not be difficult to imagine that arrestees filling out the survey would answer dishonestly. If the algorithm is configured assuming a distribution based on honest answers, this reactivity behavior will alter the risk assessment tool itself, in a way that must be accounted for. While this is a relatively straightforward example, there are many different ways technologies that inspire reactivity may destabilize existing values, incentives, and structures to such a degree that the designed tool no longer solves the critical problems in the domain.

Finally, the heterogeneous engineer must be aware that once a technology is part of the social context, new relevant social groups can arise and radically reinterpret it, return it to a state of interpretive flexibility, and suggest new mechanisms for closure. In this way, technologies that were first developed to produce fairness can be torqued to achieve other aims, even nefarious ones. In the risk assessment context, this could occur where people have agreed that a tool should be used for decarceral purposes, but after a local election, a government could aim to use it to keep people in jail. The engineering process must include "what if" scenarios that can account for the rise and fall of relevant social actors and driving concerns should the social environment change in the context of use. We cannot completely eliminate unintended consequences, but considering key choices in a technology's development at the intersection of the concerns of a variety of social groups can go a long way toward controlling ripple effects and even detecting trouble spots in advance.

3.5 An STS Lens on the Solutionism Trap

Considering when to design

Computer science programs do not typically incentivize the social science research necessary to ensure robust system use in the world—or even to fulfill the Hippocratic oath's equivalent in engineering to "first, do no harm." However, it may be that after careful consideration of the complex sociotechnical system at play—or even following an unsuccessful implementation of a "fair" system found to reinforce political distinctions and power relations—the evident and correct conclusion is to shelve the technological fix.

Fair-ML researchers would not be alone in choosing to do so. A robust conversation in the field of human-computer interaction has also addressed such concerns, cautioning system designers that in many cases, their hammer does not make the social situation at hand into a nail [11]. It may also be that careful consideration of and engagement with social contexts concerned with the administration of justice produces insights and lessons learned for other researchers to build on, instead of pre-packaged algorithms stamped with "fairness" [20]. This does not mean that fair-ML is impossible to implement. It does, however, mean that not all problems can or should be solved with technology. In such cases, taking ourselves out of the equation as a relevant social group requires and rewards researcher humility.

4 TAKEAWAYS: WHAT FAIR-ML RESEARCHERS CAN AND SHOULD DO

In a standard computer science paper, this is where we would suggest technical solutions. But guided by our analysis, our main proposed "solution" is a focus on the process of determining where and how to apply technical solutions. Much of this work will require technical researchers to learn new skills or partner with social scientists, but no less a transformation is required. We must also become more comfortable with difficult or unresolvable tensions such as that between the usefulness and dangers of abstraction. Specifically, we propose considering the five traps, perhaps in reverse order from how we have presented them. When considering designing a new fair-ML solution, this would mean determining if a technical solution:

(1) is appropriate to the situation in the first place, which requires a nuanced understanding of the relevant social context and its politics (Solutionism);
(2) affects the social context in a predictable way such that the problem that the technology solves remains unchanged after its introduction (Ripple Effect);
(3) can appropriately handle robust understandings of social requirements such as fairness, including the need for procedurality, contextuality, and contestability (Formalism);
(4) has appropriately modeled the social and technical requirements of the actual context in which it will be deployed (Portability); and
(5) is heterogeneously framed so as to include the data and social actors relevant to the localized question of fairness (Framing).

At any point, it could be reasonable to stop and consider creating a technical solution, but only if the results of the examinations of all traps are fully spelled out and included in any resulting documentation or research publications as clear limitations to the work [33].

As an example of how this process should be enacted, we return to the case of pre-trial risk assessments. We argue that the first question that should be considered is whether developing a risk assessment is appropriate given the current societal goal of reducing pre-trial detention. Before building anything, legal and social research may be required into the likely impacts of a risk assessment on the local context and the meaning of fairness within that context. Though risk assessments have been in place as part of the criminal justice pipeline for more than a century [70], each renewed commitment to developing risk assessments should still address these questions. Researchers should compare a risk assessment proposal against not only other possible algorithmic solutions, but also the existing human processes. The proposal should also be weighed against alternate policies that obviate the need for risk assessments, such as presumptively decarcerating defendants charged with nonviolent crimes [3]. If a designer can determine that a risk assessment is still the best path forward for fairness—or that there is no way to avoid creating a risk assessment, for example because it is mandated by law (e.g. Pennsylvania in the sentencing context [2])—then our

process argues that the creation of the technology should proceed with a consideration of the remaining traps.

A full examination of the Ripple Effect Trap requires examining societal motivations and anticipating potential consequences, ideally with the help of a domain expert. For risk assessments, this means understanding that introducing the risk assessments may alter the embedded values; predicting dangerousness may lead to a focus away from other societal goals such as rehabilitation. If the tool will lead away from the desired social goals, we should once again ask whether it should be built. Also important is to attempt to account for unintended consequences by, e.g., doing pilot studies to measure and monitor the impact of the introduction of the technology into the social context. This might allow one to know before full deployment whether judges will follow risk recommendations.

Avoiding the Formalism Trap involves assessing the ability of a risk assessment to encompass the complete notions of fairness and justice that are relevant in the pretrial setting. A risk assessment designer should carefully consider if and how these aims could be satisfied and assessed within the system they are building. For example, the desire for contestability might be satisfied by a combination of creating interpretable models and instituting a formal process by which a defense attorney has the chance to contest the results with the judge, based on data errors and/or the individualized situation of their client [54, 61, 69]. Concerns about relevant social groups can be obviated by taking seriously the needs of people typically underrepresented in these processes, rather than making assumptions about their welfare [23], and by understanding the power dynamics that prevent these voices from having influence in society to begin with. This can involve working with advocacy organizations, with social scientists, or directly with the populations in question. Effort should be made by the community to avoid closure and maintain interpretive flexibility until the technologies address a wider array of concerns from various social groups. Finally, the value judgments made in building the assessment should be visible so that if shifts in social context occur it is possible to understand how to re-engineer the risk assessments.

In order to avoid the Portability Trap, researchers should next carefully consider the social context for which we are creating the tool and make sure that the assumptions built into the algorithm match the properties of that context. In the case of transferring an algorithm designed to predict good hires to the context of risk assessment this means that any assumptions built into the algorithm or the fairness definition used could render such an algorithm inappropriate to the risk assessment context. In other words, the researcher should assume that any off-the-shelf algorithm used, even if it is labeled fairness-aware, may need to be modified to work within the given social constraints or may not work for the context at all. As a concrete step, researchers should adopt the concept of user scripts to clearly describe the intended uses and limitations of their code. In fact, a number of recent papers have proposed ideas along these lines [12, 27, 37, 55, 58, 60, 71].

Finally, if creating or applying an algorithm, we should avoid the Framing Trap by adopting a sociotechnical frame. This means using our new knowledge about the practical aspects of how the social context operates, and includes working with the relevant people to understand how the humans use the system, and when human actors belong inside the black box and should be modeled as such via a heterogeneous engineering approach. The designer should be able to recognize what particular decision the fairness criteria apply to—for example, the detain-or-release decision as opposed to the dangerousness determination, as discussed above. Or alternatively, that the outcome being predicted (e.g. failure to appear, re-arrest within two years) is the same as the outcome given in the data.

By following these steps, we in the fair-ML community can learn the perspectives of the relevant parties and the hidden power structures in the social systems in which we seek to intervene. This includes becoming domain experts ourselves or working with them. While we believe that deep collaboration with domain experts is the right path forward to move fair-ML into practice, we also recognize that all researchers may not have such opportunities. Lest the perfect be the enemy of the good, we encourage even researchers who work in subfields traditionally separated from their ultimate application to consider how to incorporate the sociotechnical context more directly into their work. Whether by modeling human actions or even the simpler practice of clearly stating the contextual limitations of the results, considering the social context when designing technical solutions will lead to better—and more fair—sociotechnical systems.

ACKNOWLEDGEMENTS

Thanks to Solon Barocas, Madeleine Elish, Karen Levy, Carlos Scheidegger, Christo Wilson, and our reviewers for very helpful comments on earlier drafts. Funded in part by the NSF (IIS-1633400, IIS-1633387, and IIS-1633724) and Luminate (The Omidyar Group).

REFERENCES

[1] Mark S. Ackerman. 2000. The intellectual challenge of CSCW: The gap between social requirements and technical feasibility. *Human-Computer Interaction* 15, 2-3 (2000), 179–203.
[2] Act of Oct. 27, 2010. Pennsylvania Public Law 931, No. 95. Codified at 42 Pa. C.S. §2154.7.
[3] African American Ministers In Action, et al. 2018. The use of pre-trial "risk assessment" instruments: A shared statement of civil rights concerns. http://civilrightsdocs.info/pdf/criminal-justice/Pretrial-Risk-Assessment-Full.pdf.
[4] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In *Proc. of the 35th International Conference on Machine Learning*.
[5] Philip E. Agre. 1997. Toward a critical technical practice: Lessons learned in trying to reform AI. In *Social Science, Technical Systems, and Cooperative Work: Beyond the Great Divide*, Geoffery C. Bowker, Susan Leigh Star, Les Gasser, and William Turner (Eds.). Erlbaum, 131–157.
[6] Madeline Akrich. 1992. The de-scription of technological objects. In *Shaping Technology/Building Society*, Wiebe E Bijker and John Law (Eds.). MIT Press, 205–224.
[7] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine bias. *ProPublica* (May 23 2016).
[8] Samuel R. Bagenstos. 2006. The structural turn and the limits of antidiscrimination law. *California Law Review* 94 (2006), 1–47.
[9] Chelsea Barabas, Karthik Dinakar, Madars Virza, Joichi Ito, and Jonathan Zittrain. 2018. Interventions over Predictions: Reframing the Ethical Debate for Actuarial Risk Assessment. *Proceedings of Machine Learning Research* 81, 1–15.
[10] Stephen R. Barley. 1996. Technology as an occasion for structuring: evidence from observation of CT scanners and the social order of radiology departments. *Administrative Science Quarterly* 31 (1996), 78–108.
[11] Eric P.S. Baumer and M. Silberman. 2011. When the implication is not to design (technology). In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2271–2274.
[12] Emily M. Bender and Batya Friedman. 2018. Data statements for NLP: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics (to appear)* (2018). https://openreview.net/forum?id=By4oPeX9f
[13] Michel Callon. 1986. Some elements of a sociology of translation: Domestication of the scallops and the fishermen of St. Brieuc Bay. In *Power, Action and Belief: A New Sociology of Knowledge*, John Law (Ed.). Routeledge and Kegan Paul, 196–233.

[14] Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data* 5.2 (2017), 153–163.

[15] Angèle Christin. 2017. Algorithms in practice: Comparing web journalism and criminal justice. *Big Data & Society* 4, 2 (2017).

[16] Danielle Keats Citron. 2008. Technological due process. *Washington University Law Review* 85 (2008), 1249–1313.

[17] H.M. Collins. 1990. *Artificial experts: Social knowledge and intelligent machines (inside technology).* MIT Press.

[18] Ruth Schwartz Cowan. 1985. How the refrigerator got its hum. In *The Social Shaping of Technology*, Mackenzie and Wajcman (Eds.). McGraw Hill Education, 202–218.

[19] William Dieterich, Christina Mendoza, and Tim Brennan. 2016. COMPAS risk scales: Demonstrating accuracy equity and predictive parity. (2016). Northpoint Inc.

[20] Paul Dourish. 2006. Implications for design. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 541–550.

[21] Jessica M. Eaglin. 2017. Constructing recidivism risk. *Emory Law Journal* 67 (2017), 59–122.

[22] Wendy Nelson Espeland and Michael Sauder. 2007. Rankings and reactivity: How public measures recreate social worlds. *Amer. J. Sociology* 113, 1 (2007), 1–40.

[23] Virginia Eubanks. 2018. *Automating inequality: How high-tech tools profile, police, and punish the poor.* St. Martin's Press.

[24] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *Proc. 21st ACM KDD*. 259–268.

[25] Marion Fourcade and Kieran Healy. 2013. Classification situations: Life-chances in the neoliberal era. *Accounting, Organizations and Society* 38, 8 (2013), 559–72.

[26] Sorelle A. Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. 2016. *On the (im) possibility of fairness.* Technical Report. arXiv preprint arXiv:1609.07236.

[27] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2018. *Datasheets for datasets.* Technical Report. arXiv preprint arXiv:1803.09010.

[28] Tarleton Gillespie. 2007. *Wired shut: Copyright and the shape of digital culture.* MIT Press.

[29] Ben Green. 2018. "Fair" risk assessments: A precarious approach for criminal justice reform. In *5th Workshop on Fairness, Accountability, and Transparency in Machine Learning*.

[30] Josh Greenberg. 2008. *From betamax to Blockbuster: Video stores and the invention of movies on video.* MIT Press.

[31] Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. 2018. Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning. In *Proc. of AAAI*.

[32] Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of opportunity in supervised learning. In *NIPS'16 Proceedings of the 30th International Conference on Neural Information Processing Systems*. 3315–3323.

[33] Brent Hecht, Lauren Wilcox, Jeffrey P. Bigham, Johannes Schöning, Ehsan Hoque, Jason Ernst, Yonatan Bisk, Luigi De Russis, Lana Yarosh, Bushra Anjum, Danish Contractor, and Cathy Wu. 2018. It's time to do something: Mitigating the negative impacts of computing through a change to the peer review process. https://acm-fca.org/2018/03/29/negativeimpacts/. ACM Future of Computing Blog.

[34] Deborah Hellman. 2008. *When is discrimination wrong?* Harvard University Press.

[35] David J. Hess. 2001. Editor's introduction. In *Studying those who study us: An anthropologist in the world of artificial intelligence*. Stanford University Press, xi–xxvi.

[36] Anna Lauren Hoffmann. 2019. Where fairness fails: On data, algorithms, and the limits of antidiscrimination discourse. *Under review with Information, Communication, and Society* (2019).

[37] Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. 2018. The dataset nutrition label: A framework to drive higher quality data standards. *arXiv:1805.03677 [cs]* (May 2018). http://arxiv.org/abs/1805.03677 arXiv: 1805.03677.

[38] Hal Daumé III. 2018. *A Course in Machine Learning*. http://ciml.info.

[39] Faisal Kamiran and Toon Calders. 2009. Classifying without discriminating. In *Proc. of the IEEE International Conf. on Computer, Control and Communication*.

[40] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. *Machine Learning and Knowledge Discovery in Databases* (2012), 35–50.

[41] Abraham Kaplan. 1964. *The conduct of inquiry: Methodology for behavioural science.* Chandler.

[42] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent trade-offs in the fair determination of risk scores. In *Proc. of ITCS*.

[43] Ronald Kline and Trevor Pinch. 1996. Users as agents of technological change: The social construction of the automobile in the rural United States. *Technology and culture* 37, 4 (1996).

[44] Rob Kling. 1991. Computerization and social transformations. *Science, Technology, & Human Values* 16, 3 (1991), 342–367.

[45] Issa Kohler-Hausmann. 2019. Eddie Murphy and the dangers of counterfactual causal thinking about detecting racial discrimination. *Northwestern Law Review* 113 (2019). Forthcoming.

[46] Linda Hamilton Krieger. 1995. The content of our categories: A cognitive bias approach to discrimination and equal employment opportunity. *Stanford Law Review* 47 (1995), 1161–1248.

[47] Wayne R. LaFave. 2017. *Criminal law* (6th ed.). West Academic Publishing.

[48] Bruno Latour. 1987. *Science in action: How to follow scientists and engineers through society.* Harvard University Press.

[49] Bruno Latour. 2005. *Reassembling the social an introduction to actor-network-theory.* Oxford University Press.

[50] John Law. 1987. Technology and heterogeneous engineering: The case of Portuegese expansion. In *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology*, Wiebe E. Bijker, Thomas P. Hughes, and Trevor Pinch (Eds.). MIT Press, 111–34.

[51] Charles R. Lawrence. 1987. The id, the ego, and equal protection: Reckoning with unconscious racism. *Stanford Law Review* 39 (1987), 317–388.

[52] Lawrence Lessig. 2006. *Code 2.0.* Basic Books.

[53] Zachary C Lipton, Alexandra Chouldechova, and Julian McAuley. 2018. Does mitigating ML's impact disparity require treatment disparity? *arXiv preprint arXiv:1711.07076* (2018).

[54] Isak Mendoza and Lee A Bygrave. 2017. The right not to be subject to automated decisions based on profiling. In *EU Internet Law*. 77–98.

[55] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for reporting model preformance. In *Proceedings of ACM Conference on Fairness, Accountability and Transparency (FAT*)*.

[56] Evgeny Morozov. 2013. *To save everything, click here: Technology, solutionism, and the urge to fix problems that don't exist.* Penguin UK.

[57] Trevor J. Pinch and Wiebe E. Bijker. 1984. The social construction of facts and artefacts: Or how the sociology of science and the sociology of technology might benefit each other. *Social Studies of Science* 14, 3 (1984), 399–441.

[58] Dillon Reisman, Jason Schultz, Kate Crawford, and Meredith Whittaker. 2018. Algorithmic impact assessments: A practical framework for public agency accountability. https://ainowinstitute.org/aiareport2018.pdf.

[59] Salvatore Ruggieri, Dino Pedreschi, and Franco Turini. 2010. Data mining for discrimination discovery. *ACM Trans. on Know. Disc. Data (TKDD)* 4, 2 (2010), 9.

[60] Andrew D. Selbst. 2017. Disparate impact in big data policing. *Georgia Law Review* 52 (2017), 109–195.

[61] Andrew D. Selbst and Julia Powles. 2017. Meaningful information and the right to explanation. *International Data Privacy Law* 7, 4 (2017), 233–242.

[62] Linda J. Skitka, Kathleen L. Mosier, Mark Burdick, and Bonnie Rosenblatt. 2000. Automation bias and errors: Are crews better than individuals? *The International Journal of Aviation Psychology* 10, 1 (2000), 85–97.

[63] Luke Stark. 2018. Algorithmic psychometrics and the scalable subject. *Social Studies of Science* 48, 2 (2018), 204–231.

[64] State v. Loomis 2016. 881 N.W.2d 749 (Wisconsin).

[65] Megan T. Stevenson. 2018. Assessing risk assessment in action. *Minnesota Law Review* 103 (2018). Forthcoming.

[66] Lucy Suchman. 1987. *Plans and situated actions.* Cambridge University Press.

[67] Title VII of the Civil Rights Act of 1964, Public Law 88-352 1964. Codified at 42 U.S.C. § 2000e-2.

[68] Marie VanNostrand and Christopher T. Lowenkamp. 2013. Assessing pretrial risk without a defendant interview. https://www.arnoldfoundation.org/wp-content/uploads/2014/02/LJAF_Report_no-interview_FNL.pdf. Laura and John Arnold Foundation.

[69] Rebecca Wexler. 2018. Life, liberty, and trade secrets: Intellectual property in the criminal justice system. *Stanford Law Review* 70 (2018), 1343–1429.

[70] Benjamin Alan Wiggins. 2013. *Managing risk, managing race: racialized actuarial science in the United States, 1881-1948.* Ph.D. Dissertation. University of Minnesota.

[71] Ke Yang, Julia Stoyanovich, Abolfazl Asudeh, Bill Howe, H. V. Jagadish, and Gerome Miklau. 2018. A nutritional label for rankings. *Proceedings of 2018 International Conference on Management of Data (SIGMOD'18)* (2018), 1773–1776.

[72] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. 2017. Fairness beyond disparate treatment and disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*. 1171–1180.

[73] Michael J. Zimmer and Charles A. Sullivan. 2017. *Cases and materials on employment discrimination* (9th ed.). Wolters Kluwer.