10 Things You Should Know About Algorithmic Fairness

Insights

In addressing algorithmic fairness, keep in mind:

- Stories about causation are often oversimplified.
- There is a big gap between idealized process and problems in practice.
- Managing expertise is a key organizational challenge.
- Algorithmic fairness risk is unevenly distributed.
- Think contestability and recourse, not just explainability and interpretability.
- Solutions at the individual level don't solve the problem.
- Algorithmic fairness is a wicked problem.

Algorithmic systems increasingly shape the information that people are exposed to, as well as influence their chances in areas such as employment, education, and finance. In some cases, algorithmic systems are more or less favorable to certain groups or individuals, reflecting or amplifying human biases or introducing new biases. Researchers and the press have drawn attention to issues in areas such as predictive policing, child welfare, the online housing marketplace, and image search results (for one summary of such issues see [1]), sparking substantial discussion of algorithmic (un)fairness in public policy circles, academia, and the press. Efforts are now under way to understand and prevent unjust or prejudicial

treatment of people related to race, income, sexual orientation, gender, and other characteristics historically associated with discrimination and marginalization, when and where they manifest in algorithmic systems or algorithmically aided decision making [2]. The field is evolving rapidly as stakeholders attempt to untangle these complex problems, with divergent perspectives on how best to proceed.

As someone working on algorithmic fairness at a technology company, I am glad to see the attention these issues are receiving. However, I often come across understandable misperceptions that seem at odds with the real-world practice of tackling these problems. Building on

THE CYNEFIN FRAMEWORK

The Cynefin framework created at IBM [11] has been applied in a wide range of contexts, from counterterrorism to pharmaceutical companies, and I find it extends well to algorithmic fairness. The framework organizes issues into four quadrants (Table 1).

Starting in the lower right, *obvious contexts* “are characterized by stability and clear cause-and-effect relationships that are easily discernible by everyone ... [and often] the right answer is self-evident and undisputed.” Domains that don’t change very much and highly proceduralized activities, for example, loan-payment processing, are often in this quadrant.

In contrast with obvious contexts, *complicated contexts* often have multiple right answers, usually only discernible by expert analysis that takes time but yields definite, clear results.

In *complex contexts*, the nature of the problems and the answers is less clear. The authors offer the example of a rainforest which “is in constant flux—a species becomes extinct, weather patterns change, an agricultural project reroutes a water source—and the whole is far more than the sum of its parts.... In [complex domains], we can understand why things happen only in retrospect” so we need to be patient and thoughtful while patterns gradually emerge. Experts are even more important in these complex contexts, to search for questions and answers, and identify patterns.

Finally, in *chaotic contexts*, situations are highly turbulent and there aren’t manageable patterns. The goal in these situations is usually to transform the chaotic situation into a complex one as quickly as possible.

Bringing this back to algorithmic fairness, a lot of what we currently see is in the complex quadrant. We’re in early stages, and there are a lot of things we simply don’t know, but we can expect to systematize knowledge gradually over time. As we see problems and come to understand them, we can move them into the complicated quadrant, and then eventually develop practices that are sufficiently crisp to move into the obvious quadrant. So it’s a pipeline, where we have new challenges continually coming into the upper left, and gradually moving to the lower right.

Complex “Unknown unknowns” Emergent practice	Complicated Multiple right answers, requires expertise Good practice
Chaotic No manageable patterns, turbulent Novel practice	Obvious [12] Clear right answer, known to everyone Best practice

Table 1. Summary of the Cynefin framework [11], which characterizes issues that face leaders and organizations.

the practice of sharing insights from industrial settings (e.g., [3]), in this article I will share my perspective on the nature of the problem space. My hope is that a richer understanding of on-the-ground practicalities may help those working in this area target high-leverage problems and maximize the effectiveness of their efforts to identify, remediate, and prevent potential unfairness.

1. Stories about causation are often oversimplified. You frequently

hear in the media and elsewhere that algorithmic unfairness stems from causes such as lack of hiring diversity (there weren’t any/enough people on the team who were sensitive to these issues) or from oversights in gathering training data (certain groups weren’t sufficiently represented). While these considerations are unquestionably important and must be addressed, these explanations are often incomplete and do not always provide a full or accurate account of the most salient causes. The behind-the-scenes

It’s important to recognize that unlike many other kinds of problems, many algorithmic fairness issues can’t easily be foreseen or detected before launch.

stories are usually complicated, multifaceted, and often idiosyncratic, as issues can stem from complex bidding effects, subtle biases of human

raters, coordinated attacks, or many other causes (see, e.g., [4]).

For example, the system may already be well-trained on a diverse dataset and highly accurate, and the error may be a subtle, one-in-a-million event. Once the error is discovered, the real question may not be about improving accuracy, but rather about whether the service should be turned off entirely to avoid rare but highly problematic errors. Methodologically, case-based research, incident analysis, and similar tools can help get to the heart of the issue.

2. Big gap between idealized process and problems in practice.

I often see algorithmic fairness framed in terms of idealized steps in a machine learning process (select training data, select a model, choose

model variables, etc.). In this framing, the solution is to ensure that each of the individual steps is performed fairly. However, many issues such as complex system interactions and broader ethical questions often fall

beyond the scope of those individual steps, and in fact often beyond the scope of engineering (a similar point is raised in [5]). Framing around an idealized machine learning process not only misses key issues but also has problematic implications for metrics, testing, and process improvements, which leads to my third point.

3. No one-size-fits-all solution, checklist, or test. Checklists, automated testing, and similar tools are too frequently envisioned as a primary solution to algorithmic fairness. Discussions of such tools have tended to break out along

disciplinary lines, with engineers often calling for a clear, specific, comprehensive yet simple process they can follow to quickly identify and “solve” algorithmic unfairness, and social scientists often arguing that such approaches are inadequate because, as with many sociotechnical topics, there is a great deal of contextual variation. I personally spend a lot of time explaining that algorithmic fairness is a challenge for which there is unlikely to be a single right question or answer, and that at the current stage of knowledge and understanding, we should be suspicious of checklists or overly systematized responses. Happily, there has been growing awareness across disciplinary lines in FAT* (one of the leading academic venues for research on algorithmic fairness) and other communities of the importance of context (see, for example, [5]) and the potential dangers of checklists. However, despite my general reservations, I do believe that over time we can increasingly discover crisp guidance in certain areas (see sidebar).

4. Managing expertise is a key organizational challenge. The previous point shows us that managing expertise is a central challenge for an organization that seeks to address algorithmic fairness. You need to recruit, develop, and retain experts—and not many experts exist yet because it’s so early. You also need to integrate them into the product design cycle at the right times and empower them to negotiate with product teams. In addition, you must give them time to reflect on their experiences and capture knowledge so it can be shared and ultimately systematized. From an organizational perspective, this all takes a lot of patience and discipline.

Sathya Sundaramurthy and his colleagues have relevant insights about these types of organizational challenges [6]. They conducted extensive participant observation working as analysts in security operations centers, developing valuable guidance about managing experts. For example, the security operations centers handle a constant stream of new attacks, and it’s useful, rewarding, and exciting for experts to work on these new problems. But once the experts figure out a solution, they can get burned out handling repetitive incidents of the same problem.

Accordingly, the experts need to be given sufficient time to capture and proceduralize solutions that they can hand off to *non*-experts, so they can move on to new challenges, thereby maximizing use of their expertise and avoiding burnout.

5. Algorithmic fairness risk is unevenly distributed. I want to point out another aspect of algorithmic fairness that speaks to expertise-based approaches versus other approaches.

Checklists, automated testing, and process-based approaches often presume a more comprehensive or uniform approach, where you have a set of things you’re looking for and you’re going to look for all of those things in all products—for example, you’re going to look at training-data diversity or model variables in all products. The problem with this is

that risk is not evenly distributed across specific issues within a given product, and it’s not evenly distributed across all products. Rather, a small number of products carry the bulk of the risk, and within a given product, a couple of specific issues probably carry the bulk of the risk. So if you assume that there is a cost to investigating issues (which is mostly true right now), the goal is not to test everything in a vast combinatorial space, but rather to identify and test the most risky things, the things that are most likely to cause the greatest harm. Identifying these risky things is a key skill that experts bring to the table and can gradually systematize over time.

6. Many issues are hard to foresee.

I’m often asked when algorithmic fairness should be considered in the product development cycle. The typical intuition is that, as with related issues like privacy, you want to integrate algorithmic fairness fully into the product development cycle, the earlier the better. I generally agree with this. But I think it’s also important to recognize that unlike many other kinds of problems, many algorithmic fairness issues can’t easily be foreseen or detected before launch, for example because they emerge only once the system is launched at scale, or they emerge over time while the product

is out in the world as the social or technical context changes. So it's important to think about the likely time horizon for the issues you're concerned with and leverage a couple of useful mechanisms, both of which have

potential to address needs underscored in research on industry teams' challenges in addressing fairness [1].

First, I believe there should be more emphasis on post-launch statistical testing for algorithmic fairness, about looking for signals or undesirable changes once a system has launched, in order to monitor and proactively identify problematic changes in system behavior.

Second, for certain types of problems, adversarial testing is underutilized but very useful, both pre- and post-launch. Consider, for example, journalistic-style investigation, where testers think creatively about potential failure modes and design specific inputs to an algorithmic system to see if they can elicit problematic or undesirable results. Nicholas Diakopoulos has conducted valuable research on journalistic methods for adversarial testing [7], and further research on best practices would be highly beneficial.

7. The actual distribution of problems may be different from what we've seen so far. Not only are many

of these issues hard to foresee (at least right now), there is also likely a significant selection bias in the problems we know about. We often hear about the types of issues that can be readily detected by individual users and go viral when they're posted on social media. While these sensational cases are important and motivating, they may be just the tip of the iceberg. Small systemic biases are harder to detect, and although some researchers and journalists have reported helpful examples, we probably don't have a good sense of the full space and distribution of possible problems. As we learn to do better testing, we may see different kinds of issues entirely.

8. Think contestability and recourse, not just explainability and interpretability.

Significant concerns have been raised that machine learning and more broadly algorithmic systems can be opaque, making it difficult for humans to comprehend their decisions or the implications of their operation.

Accordingly, explainability and interpretability (efforts to develop human-understandable descriptions of machine learning systems) have become active research topics in computer science, mathematics, and other fields, as well as garnering

significant attention from legal scholars, advocacy groups, and regulators, in the hope that these techniques will reveal problematic algorithmic behavior and provide a mechanism to hold algorithmic systems and those associated with them accountable for undesirable effects. While explainability and interpretability have significant potential upside, there are a lot of open

questions regarding their viability. The extent to which explainability and interpretability are achievable, particularly with existing machine learning technologies, is a matter of open debate. Further, they may come with costs; for example, systems that are more understandable may be more vulnerable to manipulation and hacking.

However, in the algorithmic fairness space, explainability and interpretability largely appear to be a means to an end rather than an end goal, and alternative properties may be more useful. In my experience, stakeholders asking for explainability and interpretability in the pursuit of algorithmic fairness are actually often seeking related properties that have recently been articulated in FAT* and related communities, such as *contestability* [8] (the ability to challenge decisions made by machine learning systems) or *recourse* [9] (the ability to change the decision of the model via actionable changes to input variables). There is an emergent understanding that different properties meet different needs. For example, a plaintiff in a lawsuit may need something very different from a doctor using a system to inform a medical diagnosis. However, there is not yet a clear understanding of which properties are needed by which

The extent to which explainability and interpretability are achievable, particularly with existing machine learning technologies, is a matter of open debate.

stakeholders at which times and for which purposes, and user research elucidating these needs would greatly advance the field.

9. Solutions at the individual level don't solve the problem.

It's been observed that the CHI community tends toward individual self-determinism, toward educating and motivating individuals to take action for their own good or for the collective good. For example, in the privacy domain, there's been an emphasis on solutions to empower and encourage users to manage their own privacy, although Aleecia McDonald and Lorrie Cranor supplied early influential evidence on the limitations of this approach [10] and concerns have increased since then. By and large, algorithmic fairness is a societal-scale issue that requires societal-scale solutions, and I'm wary of placing too much burden on affected

parties or individual members of the public. We should draw on what we've learned from other societal-scale problems like privacy and sustainability, and focus on systemic and structural effects and solutions.

10. Algorithmic fairness is a wicked problem. Many of my points boil down to this: Algorithmic fairness is a wicked problem and it needs to be treated that way, with respect for its complexity (for example, as in a recent analysis in [5]). Wicked problems, originally described in social planning, are complex societal problems, often with no clear stopping point, where attempts to address one aspect of the problem may reveal or create other

issues, and stakeholders have such different perspectives that they don't agree on what the question is, let alone what a solution would look like,

or who bears responsibility for the solution. Algorithmic fairness is such a problem, and thinking about it that way better positions us to address it.

CONCLUSION

As these issues demonstrate, algorithmic fairness is an incredibly tough, fascinating problem, with myriad social, technical, and organizational challenges. Those who work on human-computer interaction, interaction design, and related disciplines are particularly well-positioned to work across boundaries to unpack hard, messy problems, and I hope to see readers of *Interactions* make significant contributions in this area.

ACKNOWLEDGMENTS

The material in this article was originally presented as a keynote at HCIC 2018 and then at the Stanford HCI Seminar. I am grateful to both audiences for their thoughtful questions and comments on this work. My perspective on algorithmic fairness has been greatly informed by wonderful conversations and collaboration with members of Google's fairness in machine learning effort and others at Google (especially Parker Barnes, Alex Beutel, Ed Chi, Charina Chou, Sunny Consolvo, Tulsee Doshi, Sarah Fox, Jason Freidenfelds, Jen Gennai, Yoni Halpern, Ben Hutchinson, Patrick Gage Kelley, Lea Kissner, Meg Mitchell, Dan Russell, Andy Schou, Andrew Smart, Ian Wilbur, and

Lawrence You), as well as members of the Algorithmic Fairness & Opacity working Group (AFOG) at UC Berkeley (particularly Jenna Burrell, Daniel Kluttz, Joshua Kroll, and Deirdre Mulligan).

ENDNOTES

1. Holstein, K., Wortman Vaughan, J., Daumé III, H., Dudík, M., and Wallach, H.. Improving fairness in machine learning systems: What do industry practitioners need? *Proc. CHI 2019*. ACM, New York, 2019.
2. While there are multiple possible definitions of algorithmic fairness, this interpretation is well-aligned with commonly taken social justice perspectives on the issue.
3. Cramer, H., Garcia-Gathright, J., Springer, A., and Reddy, S. Assessing and addressing algorithmic bias in practice. *Interactions* 25, 6 (2018), 58–63.
4. Zunger, Y. Asking the right questions about AI. Medium. Oct. 11, 2017; <https://medium.com/@yonatanzunger/asking-the-right-questions-about-ai-7ed2d9820c48>
5. Selbst, A.D., boyd, d., Friedler, S.A., Venkatasubramanian, S., and Vertesi, J. Fairness and abstraction in sociotechnical systems. *Proc. FAT* 2019*. ACM, New York, 2019.
6. Sundaramurthy, S.C., Bardas, A.G., Case, J., Ou, X., Wesch, M., McHugh, J., and Rajagopalan, S.R. A human capital model for mitigating security analyst burnout. *Proc. SOUPS 2015*. USENIX Assoc., Berkeley, CA, 2015.
7. Diakopoulos, N. Algorithmic accountability: Journalistic investigation of computational power structures. *Digital Journalism* 3, 3 (2015), 398–415.
8. Mulligan, D. Beyond algorithmic scapegoating: Fostering cultures of algorithmic responsibility through administrative law and design. *FAT* 2019 Keynote*.
9. Ustun, B., Spangher, A., and Liu, Y. Actionable recourse in linear classification. *Proc. FAT* 2019*. ACM, New York, 2019.
10. McDonald, A.M. and Cranor, L.F. The cost of reading privacy policies. *I/S: A Journal of Law and Policy for the Information Society* 4, 3 (2008), 543–568.
11. Snowden, D.J. and Boone, M.E. A leader's framework for decision making. *Harvard Business Review* (Nov. 2007), 69–76.
12. This quadrant was originally named “simple” but subsequently renamed “obvious.”

Allison Woodruff is a senior staff user experience researcher at Google. She co-founded Google's fairness in machine learning effort, and also worked on Google's privacy team. Her current research at Google focuses on algorithmic fairness as well as AI ethics. → woodruff@acm.org