ASSESSING AND ADDRESSING ALGORITHMIC BIAS IN PRACTICE

Henriette Cramer, Jean Garcia-Gathright, Aaron Springer, and **Sravana Reddy,** Spotify

Insights

- There are no ready-made, industry-standard processes to address algorithmic bias in practice.
- Barriers include understanding actual issues and their concrete impact, finding applicable solutions, and organizational challenges to at-scale implementation.
- It is necessary to translate research literature into pragmatic processes.

Unfair algorithmic biases and unintended negative side effects of machine learning (ML) are gaining attention—and rightly so. We know that machine translation systems trained on existing historical data can make unfortunate errors that perpetuate gender stereotypes. We know that voice devices underperform for users with certain accents, since speech recognition isn't necessarily trained on all speaker dialects. We know that recommendations can amplify existing inequalities. However, pragmatic challenges stand in the way of practitioners committed to addressing these issues. There are no clear guidelines or industry-standard processes that can be readily applied in practice on what biases to assess or how to address them. While researchers have begun to create a rich discourse in this space, the translation of research discussions into practice is challenging. Barriers to action are threefold: understanding the issues, inventing approaches to address the issues, and confronting organizational/institutional challenges to implementing solutions at scale.

In this article, we describe our engagement with translating research literature into processes

IMAGE BY LIUZISHAN

within our own organization as a start to assessing both intended data and algorithm characteristics and unintended unfair biases. This includes a checklist of questions teams can ask while making decisions in data collection, modeling decisions, and assessing outcomes, as well as a product case study involving voice. We discuss lessons learned along the way and offer suggestions for others seeking guidelines and recommendations for best practices for the mitigation of algorithmic bias and negative effects of ML in consumer-facing product development.

TRANSLATING RESEARCH INTO PRAGMATIC DECISION SUPPORT

Over the past several years, machine learning (ML) has developed into a ubiquitous and powerful tool. By finding patterns in thousands or millions of training examples, ML systems help us keep our inboxes tidy by identifying spam, provide personalized recommendations for content to consume and buy, and even analyze medical images to potentially help doctors detect cancer. *Bias* in these contexts refers to an inherent skew in a model that causes it to underperform in a systematic way. In research on algorithmic fairness, bias is sometimes defined as unfair discrimination: negative impacts of ML efforts that unfairly (dis)favor particular groups or individuals. Alternatively, it can be framed as the characteristics—the biases—a system and its data have, some intended and some unintended. In the latter interpretation, all data and algorithms have biases. Curated datasets are always an approximation to the true underlying distribution. The challenge for product teams then is to make decisions that lead to intended behavior, and both foresee and counter unintended consequences. Research is not enough to make this happen in practice; we need to bridge the gap between the academic literature and product decisions.

While general awareness of the power and reach of ML, the reliance of ML on data, and the possibility of bias has increased recently in the media, these issues have been discussed for over 20 years within HCI and related areas. Batya Friedman and Helen Nissenbaum [1] presented a prescient taxonomy of biases in computational systems in 1996. Since then, the body of literature has been steadily growing. Research communities spanning industry and academia, such as FATML [2], have matured into dedicated FAT* venues [3]. Microsoft, Google, and Facebook have all announced research into algorithmic bias. Initiatives such as AI Now [4] are gaining deserved attention with guidelines for algorithmic impact assessment. Research organizations like Data & Society [5] and mainstays like the World Wide Web Foundation [6] and ACM are presenting accountability primers and helpful principles. In the wake of Europe's cookie and GDPR legislation, the unforeseen consequences of machine learning have also received political attention.

A proactive approach is important, but it can be difficult to operationalize literature in step-by-step processes and reconcile methods with gritty on-the-ground demands. As ML becomes more ubiquitous, guidelines for practitioners such as Gebru et al.'s datasheets [7] become increasingly relevant. Developing processes to assess, and certainly to address, issues within organization-specific contexts still requires work. Product teams who own data-driven platforms or features are continuously faced with decisions about data collection, maintenance, and modeling. In order for algorithmic-bias efforts to succeed, teams must have support for making these decisions thoughtfully and intentionally. It is also crucial to assess data quality and understand the appropriate measures of effectiveness. At least three complementary types of effort are required (Figure 1). The first is *research and analysis* to know how to assess and address bias. This involves translating both existing research into the organizational context, as well as case studies into specific products. The second is developing *processes* that are easy to integrate into existing product cycles. This requires organizational work: education and organizational coordination. The third is engaging with *external communities* to exchange lessons learned and ensure that the work done internally keeps up with the state of the art. Each of these come with specific challenges that have to be addressed within the product, technical, and organizational contexts.

A proactive approach is important, but it can be difficult to operationalize literature in step-by-step processes and reconcile methods with gritty on-the-ground demands.

PILOTING GENERAL AND FEATURE-SPECIFIC METHODS

Teams need concrete recommendations and methods that fit specific contexts. This includes general methods or checklists that are shared across organizations, and application- or modality-specific methods (Figure 2). Checklists provide a shared framework for how to approach and communicate about data and outcome characteristics, and provide the foundation for a structured approach across teams.

General methods: from research to checklist. Very human decisions affect machine-learning outcomes. A team's success criteria—which data to collect, which models to use, and which people to involve in quality assessment—can all be steered but are not always fully planned out. To allow teams to make the right decisions, they must know what questions to ask. In our case, we evaluated a selection of existing bias frameworks (including [1,8,9]) and translated them into an easy-to-digest summary of bias types, while adding

team composition and team expertise as additional categories.

Taxonomies of existing biases can be very helpful here but must be turned into pragmatic support for teams. Certain types of taxonomies are easier than others in this regard. For example, when we reviewed Friedman and Nissenbaum's taxonomy of biases in computational systems, it included categories such as preexisting bias, technical bias, and emergent bias. While this work was informative, it was difficult to use the taxonomy in practice, as the categories did not point to underlying causes to evaluate or actions to take. More recent taxonomies of algorithmic and data biases allowed us to classify problems in a way that points out how to intervene.

For data bias, we used Olteanu et al.'s perspective on systemic distortions that compromise data's representativeness [8] as a starting point. The framework comprehensively examines biases introduced at different levels of data gathering and usage. While it originally focuses on social data analysis, we were able to translate it for our purposes. It also raises an interesting dilemma, as representative data may reflect existing societal biases and existing disadvantages. Similarly, the bias taxonomy by Ricardo Baeza-Yates [9] was relatively easy to translate. It consists of six types: activity bias, data bias, sampling bias, algorithm bias, interface bias, and self-selection bias. These biases form a directed cycle graph; each step feeds biased data into the next stage, where additional biases are introduced. The model's breakdown potentially makes it easier to find targets for initial intervention.

We summarized three categories of entry points for biases (Figure 3):

- *Data*: characteristics of the input data
- *Algorithm and team*: model characteristics as well as team decisions
- *Desired outcomes*, such as recommendation content and served populations.

The checklist asks whether each identified bias is expected to affect the project's results or its priority, and what could be done to address issues.

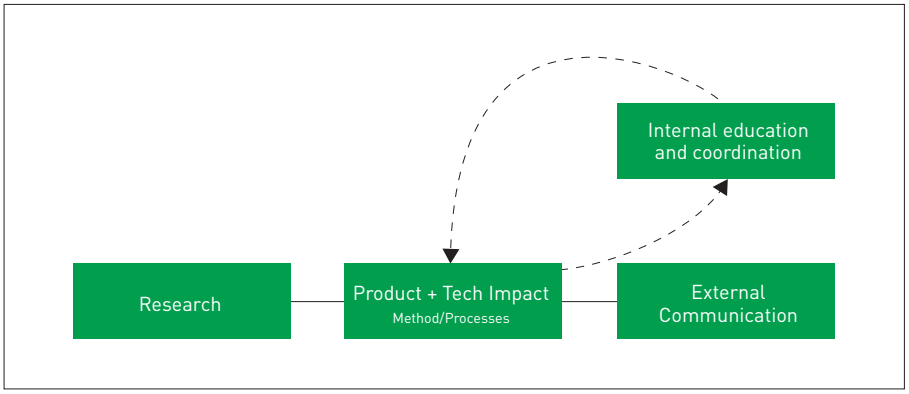

Figure 1. Three types of effort required to address algorithmic bias.

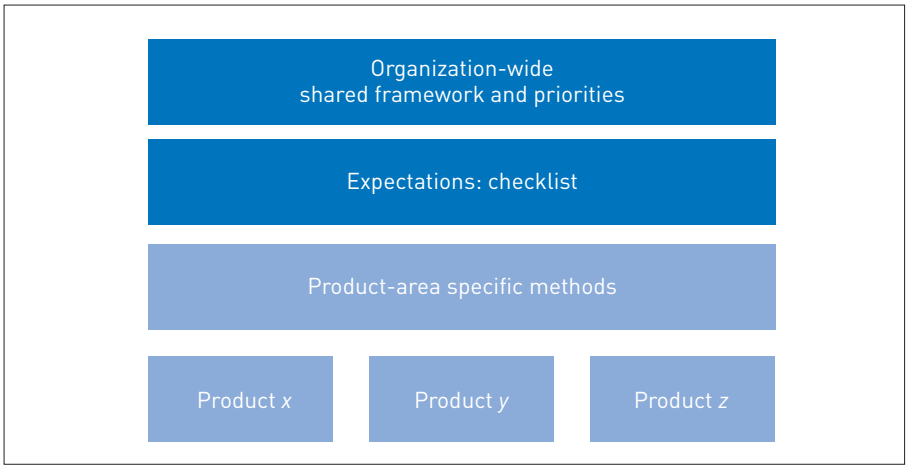

Figure 2. Organization-wide and product-specific methods.

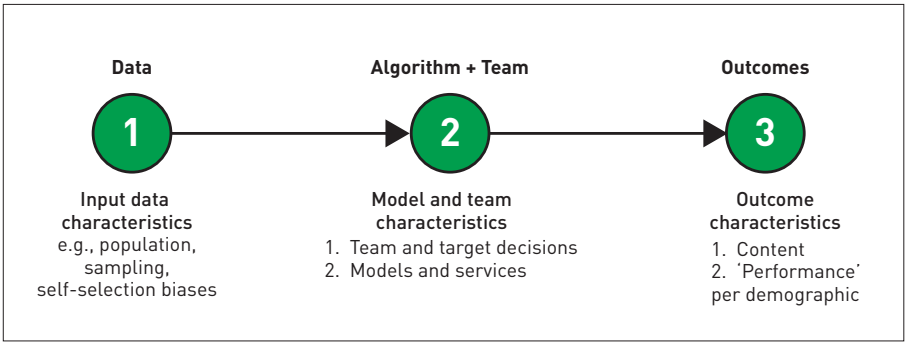

Figure 3. Three categories of bias entry points.

While such checklists can never be a complete overview, they can support prioritization and education.

Checklist lessons learned. During the early pilot with our v0 checklist with two different machine-learning-heavy teams, data quality, especially the differing availability of historical data for different markets, and the cyclical nature of bias were reported as potential issues. The effect of organizational structure also became apparent in the pilot. There could be biases in upstream datasets, but it was not directly apparent whether these datasets could be changed, and who should then take on that task—especially when side effects could be unclear. When services and pipelines build on each other, alignment is necessary across different teams to make sure resourcing is most effective, especially when infrastructure has to be built. Teams own different parts of the infrastructure. A change in one pipeline may affect multiple services, with potentially unforeseen consequences for products in the wild. Teams may also need help negotiating with other teams' priorities when they find out they cannot fix an issue themselves, or when they may be affecting outcomes of another team's

product. Getting organized using a shared framework becomes even more important to be able to deal with such cases.

From checklist to case study: voice. Beyond general methods, we also need methods for specific types of products and specific issues we encounter in practice. One such example is voice applications. While voice isn't new, it is relatively new to be used at scale in diverse applications. In specialized and creative domains, this leads to issues. For example, in music, artists may name themselves with symbols (e.g., M△S▲C△RA) that most standard automated speech recognition (ASR) systems cannot transcribe. Some have alternative spellings, such as the track "Hot in Herre" or the artist 6LACK, pronounced "black." Code switching between languages is also common, but most ASR systems are trained on data from one main locale. Especially when wide developer communities start to rely on standard, not domain-specific ASR APIs trained on large-scale data from other application types, this can have serious consequences. Certain types of content become much less accessible (Figure 4).

To get a firmer grasp on potential issues, we developed a way to detect such content at scale by detecting unexpected differences in content engagement, or popularity, between modalities [10]. Categorization of underserved content resulted in a typology of content types of linguistics practices that can make content harder to surface. For music, we found that genres such as hip-hop and country were disproportionately affected, as creative language usage and code switching between languages are a valued part of these subcultures. After detection, it is possible to correct these issues by collecting multiple pronunciations from users without pre-knowledge, or via crowdsourcing, and have these transcribed. Artists' intended pronunciation and spelling may not match those of users, crowdworkers, or the ASR transcription. Crowdsourced pronunciations are thus not guaranteed to include the right pronunciation per se, but they do provide a scalable and relatively easy way to correct some inaccessible content.

More complex models would have been possible to address each of the 11 types of issues we found. However, these would require more resources and changes to infrastructure. This points to the need to value not just the most complex model literature, but also easy-to-apply solutions and sociotechnical work to understand team challenges.

IMPACTING PRODUCTS: ORGANIZATIONAL COMMUNICATION AND EDUCATION

Developing methods to address biases is not enough: Considerable organizational work may have to be done as well. While cultural changes are needed to address biases early on, a deep understanding of the challenges that teams face in their day-to-day practice will help make algorithmic-bias efforts more successful. This includes developing lightweight tools that support decision making, ensuring that methods fit rapid-delivery engineering practices, and breaking down the daunting task of addressing algorithmic bias into smaller pieces. It also includes a conscious effort to educate and to iterate on potential tools provided to teams. In larger organizations, it means both resourcing research efforts as well as evangelization and supporting teams in their specific questions.

Aligning on priorities and interdependencies. Teams have competing demands and must deal with changing circumstances. In agile contexts, teams may be self-organizing and cross-functional. Continual learning and changing requirements based on new insights also means continual change. Weighing which approaches would be most helpful and deciding how to prioritize between different issues can benefit from decision support. Organizations have to agree on what types of demographics, biases, and stakeholders to consider in their algorithmic-bias efforts. In music contexts, genres and local music cultures are important considerations, while they may be less relevant in other contexts. Trade-offs between different stakeholders, or even different user demographics, may be necessary; it may not be obvious what the right outcome is.

Communicate the minimal viable product steps. Algorithmic bias to a certain extent can be seen as technical debt. Bias is much easier to tackle when working with a new product rather than one that has been running for a while, or where a variety of models are working together. Unintended biases are self-reinforcing, recursive, and much harder to eliminate if ignored at the beginning. However, in the early stages of product development, as well as in the development of startups, scaling is not an option. Target users may change over time. Datasets may be limited by necessity, as no user data is available. This also means that the data necessary to even assess algorithmic bias and which demographics would need additional attention is often not available. Quality evaluations may be dependent on the knowledge available at that early time. Prioritizing such speculative endeavors can be at odds with agile development and its rapid delivery toward specific user stories and maximum impact with minimal investment. Addressing algorithmic bias in product development thus may require short-term narrow steps, with continual improvement paths forward. It may also require taking time rather than "moving fast and breaking things."

Education. Structuring algorithmic-bias assessment methods in a manner that fits into prioritization processes in teams may be more fruitful than an approach that is presented as extra work. This

Unintended biases are self-reinforcing, recursive, and much harder to eliminate if ignored at the beginning.

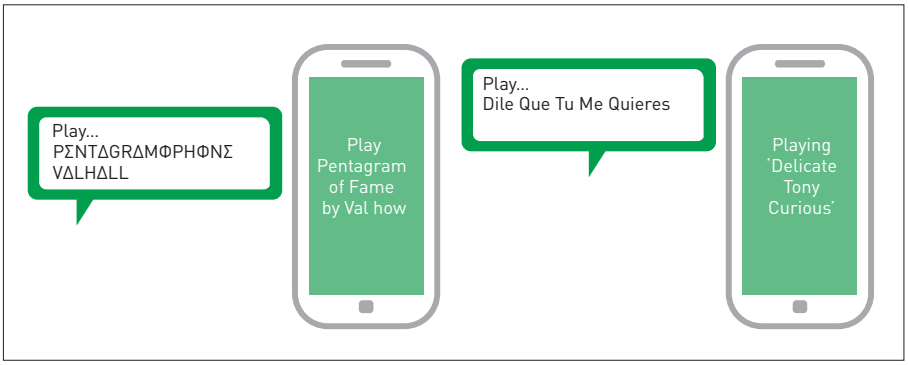

Figure 4. Example of voice-recognition issues within the music domain.

means that assessing and addressing bias needs to be part of the product team's normal goal-setting processes, while in-depth research potentially has to be resourced separately so it does not weigh on teams' local roadmaps. Education efforts have to be structured in a manner that doesn't appear as yet another training. In our pilot, we started to include algorithmic bias in existing machine-learning courses for different internal audiences, and in both engineering and diversity events, rather than requiring special attendance.

Companies can highlight the importance of algorithmic-bias efforts by including it in company-wide baseline expectations combined with specific individual team goals, by resourcing the simplification of tools to fit company-specific practices, and by making (future) dashboards and meta-datasets widely available. This does benefit from leadership buy-in, which requires quick executive summaries on *why* algorithmic bias is important and *how* to best address it organizationally. Different arguments will convince different audiences, and a step-by-step approach is crucial, rather than a purist approach that strives to solve everything all at once or that potentially punishes developers for sharing issues that they encounter.

INCREASING IMPACT AS A RESEARCH COMMUNITY

Great work is being done in the HCI and machine-learning communities to address algorithmic bias; however, in practice it's challenging to consume all of it and even harder to translate the discussions and findings into practical approaches for product teams. Continually following the scientific literature on fairness requires specific expertise and focus not always available in production settings. Turning research literature into guidelines applicable in practice requires a translation that cannot be expected from all paper authors. Applied industry researchers absolutely must be able to fulfill this mediating role, hopefully aided by more accessible literature.

Easy-to-use summaries of papers and the informal sharing of concrete experiences with both successful and problematic results is useful in better understanding what gaps to close, and how to best have impact. For example, when inevitable discussions on "what fair means" arise, summaries such as the *21 Definitions of Fairness* [11] are invaluable points of reference. When setting up an effort to assess and address bias, overviews of how other companies are approaching the issue are especially helpful and can motivate prioritization and resourcing. This also means that we need to be open to sharing less-than-ideal learning experiences. In the majority of cases, teams want to make the right decisions but don't necessarily know how—let's help make this easier. In this article, we have offered insights into our approach to doing so.

Endnotes

1. Friedman, B. and Nissenbaum, H. Bias in computer systems. *ACM Trans. Inf. Syst. 14*, 3 (1996), 330–347.
2. FATML; https://www.fatml.org/resources/principles-for-accountable-algorithms
3. ACM Conference on Fairness, Accountability, and Transparency (ACM FAT*); https://fatconference.org
4. AI Now institute. Algorithmic impact assessments: A practical framework for public agency accountability. Apr. 2018; https://ainowinstitute.org/aiareport2018.pdf
5. Data & Society. Algorithmic accountability primer. Apr. 2018; https://datasociety.net/wp-content/uploads/2018/04/Data_Society_Algorithmic_Accountability_Primer_FINAL-4.pdf
6. World Wide Web Foundation. Algorithmic accountability report, 2017; https://webfoundation.org/docs/2017/07/Algorithms_Report_WF.pdf
7. Gebru, T., Morgenstern, J., Vecchione, B., Wortman Vaughan, J., Wallach, H., Daumeé III, H., and Crawford, K. Datasheets for datasets. 2018; https://arxiv.org/abs/1803.09010
8. Olteanu, A., Castillo, C., Diaz, F., and Kiciman, E. Social data: Biases, methodological pitfalls, and ethical boundaries. 2016; http://dx.doi.org/10.2139/ssrn.2886526
9. Baeza-Yates, R. Data and algorithmic bias in the web. *Proc. of the 8th ACM Conference on Web Science*. ACM, New York, 2016; https://doi.org/10.1145/2908131.2908135
10. Springer, A. and Cramer, H. "Play PRBLMS": Identifying and correcting less accessible content in voice interfaces. *Proc. of CHI '18*. ACM, New York, 2018.
11. Narayanan, A. FAT* 2018 tutorial: 21 fairness definitions and their politics.

Henriette Cramer is a lab lead at Spotify, where her research has focused on voice interactions and road managing Spotify's algorithmic accountability effort. She is especially interested in data and design decisions that affect machine-learning outcomes, and the (mis)match between machine models and people's perceptions.
→ henriette@spotify.com

Jean Garcia-Gathright is a machine-learning engineer and researcher at Spotify, where she specializes in the evaluation of data-driven models that power engaging, personalized user experiences.
→ jean@spotify.com

Aaron Springer is a Ph.D. candidate at University of California Santa Cruz. His research focuses on the user experience of machine learning, including fairness, trust, and transparency.
→ alspring@ucsc.edu

Sravana Reddy is a researcher at Spotify working in natural language processing and machine learning. Her interests lie in computational sociolinguistics, NLP for creative language, audio processing, and of late, fairness and bias in ML. She received her Ph.D. from the University of Chicago.
→ sravana@spotify.com

DOI: 10.1145/3278156