

Algorithms as culture: Some tactics for the ethnography of algorithmic systems

Big Data & Society
July–December 2017: 1–12
© The Author(s) 2017
DOI: 10.1177/2053951717738104
journals.sagepub.com/home/bds



Nick Seaver

Abstract

This article responds to recent debates in critical algorithm studies about the significance of the term “algorithm.” Where some have suggested that critical scholars should align their use of the term with its common definition in professional computer science, I argue that we should instead approach algorithms as “multiples”—unstable objects that are enacted through the varied practices that people use to engage with them, including the practices of “outsider” researchers. This approach builds on the work of Laura Devendorf, Elizabeth Goodman, and Annemarie Mol. Different ways of enacting algorithms foreground certain issues while occluding others: computer scientists enact algorithms as conceptual objects indifferent to implementation details, while calls for accountability enact algorithms as closed boxes to be opened. I propose that critical researchers might seek to enact algorithms ethnographically, seeing them as heterogeneous and diffuse sociotechnical systems, rather than rigidly constrained and procedural formulas. To do so, I suggest thinking of algorithms not “in” culture, as the event occasioning this essay was titled, but “as” culture: part of broad patterns of meaning and practice that can be engaged with empirically. I offer a set of practical tactics for the ethnographic enactment of algorithmic systems, which do not depend on pinning down a singular “algorithm” or achieving “access,” but which rather work from the partial and mobile position of an outsider.

Keywords

Algorithms, ethnography, multiplicity, tactics, methodology

This article is a part of special theme on Algorithms in Culture. To see a full list of all articles in this special theme, please click here: <http://journals.sagepub.com/page/bds/collections/algorithms-in-culture>.

Terminological anxiety

At a conference on the social study of algorithms in 2013, a senior scholar stepped up to the audience microphone: “With all this talk about algorithms,” he said, “I haven’t heard anybody talk about an *actual* algorithm. Bubble sort, anyone?”¹ Over two days, speakers had covered algorithmic topics from Google’s autocomplete feature to credit scoring, but the questioner was right: they had not examined anything like bubble sort, a simple algorithm for ordering a list. Bubble sort, the questioner implied, was an *actual* algorithm; autocomplete was something else.

The conference, *Governing Algorithms*, hosted at New York University, was an early moment in the growing transdisciplinary field of critical algorithm studies—broadly speaking, the application of humanistic and

social scientific approaches to algorithms.² But already, one of the field’s central tensions was evident: Did these humanists and social scientists, taking on objects that had, until recently, been the domain of computer scientists, know what they were talking about?

As “algorithm” drifted out of computer science and into popular and critical academic discourse, it seemed to signify a renewed concern for technical specificity. Where “Big Data” was vague—originating in an overheated marketing discourse—algorithms were precise. They were the core stuff of computer science,

Tufts University, USA

Corresponding author:

Nick Seaver, Tufts University, Medford, MA 02155, USA.
Email: nick.seaver@tufts.edu



definitionally straightforward and, for many humanists, as distilled a case of rationalizing, quantifying, procedural logics as it was possible to find (see, e.g., Totaro and Ninno, 2014). The work to be done was clear: apply classic critiques of rationality, quantification, and procedure to these new objects and hit “publish.”

Yet, just as critical scholars picked them up, algorithms seemed to break apart. They had become, in the words of *Governing Algorithms*’ organizers, “somewhat of a modern myth” (Barocas et al., 2013: 1), attributed with great significance and power, but with ill-defined properties. Where an algorithm like bubble sort was so simple it could be described in a sentence,³ the charismatic mega-algorithms that had caught scholars’ attention—Google’s search, Facebook’s newsfeed, New York City’s predictive policing system, Netflix’s recommender, and so on—were more complicated and harder to specify. Although these systems were called “algorithms” by the public, academics, and even by their makers, the distance between them and their Computer Science 101 counterparts was unsettling. Maybe, in our inexperienced enthusiasm, critical algorithm studies scholars had made a mistake. Maybe we don’t understand what algorithms are, after all.

I take terminological anxiety to be one of critical algorithm studies’ defining features. But this is not because, as disciplinary outsiders, we are technically inept. Rather, it is because terminological anxieties are first and foremost anxieties about the boundaries of disciplinary jurisdiction, and critical algorithm studies is, essentially, founded in a disciplinary transgression. The boundaries of expert communities are maintained by governing the circulation and proper usage of professional argot, demarcating those who have the right to speak from those who do not (Fuller, 1991; Gal and Irvine, 1995; Gieryn, 1983), and algorithms are no different. Our worry about what “algorithm” means has more to do with our positions vis-a-vis other groups of experts than it has to do with our ability to correctly match terms and referents.

Rather than offering a “correct” definition, this article advances an approach to algorithms informed by their empirical profusion and practical existence in the wild—always at the boundaries of diverse communities of practice. It looks to anthropology for inspiration, both because my own training is anthropological and because anthropology proves useful for thinking through encounters between disparate knowledge traditions. It outlines an ethnographic approach to algorithms because ethnographic methods are well suited to the concerns that tend to occupy critical scholars—particularly concerns about how formalisms relate to culture. Ethnography roots these concerns in empirical soil, resisting arguments that threaten to wash away ordinary experience in a flood of abstraction.

Rather than entering the field with a definition in hand, I propose using fieldwork to discover what algorithms are, in practice. After exploring two competing visions of what an anthropology of algorithms might entail, I offer a set of practical tactics for the ethnography of algorithmic systems, derived from my own ethnographic experience.

Anthropology I: Algorithms in culture

A straightforward solution to our definitional crisis would be to take some expert definition as decisive: let computer scientists define “algorithms” and then examine how those things interact with our own areas of expertise. Like many straightforward solutions, this one has complications. As Paul Dourish has noted, “the limits of the term algorithm are determined by social engagements rather than by technological or material constraints” (2016: 3). That is to say, different people, in different historical moments and social situations have defined algorithms, and their salient qualities, differently. A data scientist working at Facebook in 2017, a university mathematician working on a proof in 1940, and a doctor establishing treatment procedures in 1995 may all claim, correctly, to be working on “algorithms,” but this does not mean they are talking about the same thing. An uncritical reliance on experts takes their coherence for granted and runs the risk of obscuring a key interest of critical scholars: what happens at the edges of knowledge regimes.

Given this instability and diversity, Dourish advances an anthropological case for taking on a “proper” expert definition of algorithms: we should do it not because this definition offers “a foundational truth of the nature of algorithms as natural occurrences,” but because it is what engineers do (2016: 2). In anthropological parlance, “algorithm” is an *emic* term, an actor’s term, part of the linguistic architecture through which software engineers make sense of the world and their work. Moreover, taking on such a definition facilitates critical algorithm scholars’ political projects: “Critiques of algorithmic reasoning that set their own terms of reference for key terminology,” he writes, “are unlikely to hit home” (Dourish, 2016: 2). If we want to understand engineers and get them to listen to us, then we need to use terms as they do; it is a matter “of ethnographic responsibility and practical politics” (Dourish, 2016: 2).

But what exactly is the emic definition of “algorithm,” and where should we find it? Dourish’s argument hinges on a definition of group boundaries. He writes:

When technical people get together, the person who says, “I do algorithms” is making a different statement

than the person who says, “I study software engineering” or the one who says, “I’m a data scientist” and the nature of these differences matters to any understanding of the relationship between data, algorithms, and society. (Dourish, 2016: 3)

The boundaries of the “algorithm” are social boundaries: between algorithm people and other “technical people,” and between technical people and their non-technical others, who may not understand the definitions in play. Now imagine an ethnographer encountering these technical people; if she did not know what an algorithm was supposed to be, she would have to suss it out from how they speak and act. (“*These* people say ‘algorithm’ whenever they want *those* people to stop talking...”) “*Those* people talk about algorithms like precious intellectual property, while *these* people talk about them like plumbing...”) Dourish suggests that, if an ethnographer does this, she will end up where he begins: “In computer science terms, an algorithm is an abstract, formalized description of a computational procedure” (2016: 3).

This is, however, an empirical question: Do the people who “do” algorithms today actually treat them according to this “proper” definition? Ethnography often throws analytic frameworks into disarray, and this proved true in my own fieldwork with US-based developers of algorithmic music recommender systems. After setting out to study engineers specifically, I realized that many more actors shaped the systems these companies built. Eventually, I interviewed half the employees of a recommender company that at the time employed roughly 80 people. These people’s jobs ranged from summer intern to CEO, and from systems engineer to front-end web developer. All of them, whether “technical people” or not, were party to the production of algorithms, but in the office, the “algorithm” seemed to be nowhere in particular.

So I sought it out, asking people to identify the algorithms they worked on. They typically balked at this question, and even people in the most “algorithmic” roles at the company, working on machine learning infrastructure or playlist personalization, located “the algorithm” just outside the scope of their work, somewhere in the company’s code. One, a senior software engineer with a prestigious undergraduate degree in computer science told me that her training on algorithms in theory was irrelevant to her work on algorithms in practice, because algorithms in practice were harder to precisely locate: “It’s very much black magic that goes on in there; even if you code a lot of it up, a lot of that stuff is lost on you.” The “algorithm” here was a collective product, and consequently everyone felt like an outsider to it.

When my interlocutors talked about algorithms, it was usually as part of a popular critical discourse that pitted algorithmic recommendation against human curators, claiming that “algorithms” could not understand music well enough to recommend it (e.g. Titlow, 2013). These engineers, humans within a system described as inhuman, resisted this framing: “algorithms are humans too,” one of my interlocutors put it, drawing the boundary of the algorithm around himself and his co-workers. Sitting in on ordinary practical work—whiteboarding sessions, group troubleshooting, and hackathon coding—I saw “algorithms” mutate, at times encompassing teams of people and their decisions, and at other times referring to emergent, surprisingly mysterious properties of the codebase. Only rarely did “algorithm” refer to a specific technical object like bubble sort. When pressed, many of my interlocutors could recite a proper definition, but these definitions were incidental to the everyday decision making I observed in the field. In practice, “algorithm” had a vague, “non-technical” meaning, indicating various properties of a broader “algorithmic system” (Seaver, 2013), even in nominally “technical” settings.

So, while Dourish’s argument for emic definition is sound, we cannot know those definitions in advance or assume that they will be precise and stable. If we look to the places where algorithms are made, we may not find a singular and correct sense of “algorithm.” Assuming that we will reifies a vision of the algorithm that risks obscuring humanistic concerns and blinding us to diversity in the field. “Technical people” are not the only people involved in producing the sociomaterial tangles we call “algorithms” and in practice, even they do not maintain the definitional hygiene that some critics have demanded of each other. A diverse crowd of people, using a wide array of techniques and understandings, produce the “algorithm” in a loosely coordinated confusion. Neglecting this is especially problematic for the algorithms that the public and most critics focus on: these are distributed, probabilistic, secret, continuously upgraded, and corporately produced.

Moreover, the “correct” definition of algorithms has been used precisely to isolate them from the concerns of social scientists and humanists, and it has been picked up by advocates and critics alike to set algorithmic processes apart from cultural ones. Dourish suggests that clarifying what algorithms are facilitates a discussion of how they interact with what they are not, and he provides a set of algorithmic “others” (things they are often confused with, but properly distinct from). This list makes clear how the proper definition of algorithms serves to distinguish them from typical critical concerns: algorithms are not automation (thus excluding questions of labor), they are not code (thus excluding

questions of texts), they are not architecture (thus excluding questions of infrastructure), and they are not their materializations (thus excluding questions of sociomaterial situatedness) (2016: 3–5). Such a definition carves out a non-social abstract space for algorithms, artificially setting them apart from the various concerns that they tangle with in practice. The technologist who insists that his facial recognition algorithm has no embedded politics and the critic who argues that algorithmic music recommendation is an exogenous threat to culture both rely on an a priori distinction between cultural and technical stuff.

Let's call this the *algorithms in culture* approach. It hinges on the idea that algorithms are discrete objects that may be located within cultural contexts or brought into conversation with cultural concerns. Understood as such, algorithms themselves are *not* culture. They may *shape* culture (by altering the flows of cultural material), and they may be shaped *by* culture (by embodying the biases of their creators), but this relationship is like that between a rock and the stream it is sitting in: the rock is not part of the stream, though the stream may jostle and erode it and the rock may produce ripples and eddies in the stream. In this view, algorithms can affect culture and culture can affect algorithms *because* they are distinct.

Anthropology 2: Algorithms as culture

Something like what has happened to computer scientists and the term “algorithm” happened earlier with anthropologists and “culture”: a term of art for the field drifted out of expert usage, and the experts lost control. Through the 1980s, American anthropologists were becoming generally skeptical of “culture” as an explanatory concept or object of study (Abu-Lughod, 1991). Its implicit holism and homogenizing, essentialist tendencies seemed politically problematic and ill-suited to the conflictual, changing shape of everyday life in anthropological field sites.

But while this skepticism grew, the culture concept gained purchase outside of anthropology: “cultures” were taken to mark diverse, bounded groups with timeless traditions, often synonymous with ethno-national identities; companies that once described their employees as a “family” might now say they had a “culture,” which designated an attitude toward work and, perhaps, what food and games were in the break room (Helmreich, 2001; Strathern, 1995). While anthropologists debated the usefulness or even existence of culture, it became a matter of concern among people with no obligation to anthropological definitions.

As anthropologists increasingly studied people with the power to resist outside explanations (Gusterson, 1996; Nader, 1969), this became a practical problem,

not just an epistemological one. While anthropologists could critique the use of “culture” by fellow fieldworkers, these new users of “culture” were often influential parts of the social scene anthropologists wanted to describe. Groups of people linked by employer or ethnicity might take up “culture” as a project, making their understandings influential, even if anthropologists disagreed with them; vernacular theories of culture could shape social action in their image.

Consequently, many anthropologists turned from a vision of cultures as coherent symbolic orders to practice as the stuff of cultural life (Bourdieu, 1972; Ortner, 1984). As Lila Abu-Lughod put it, the practice approach to culture “is built around problems of contradiction, misunderstanding, and misrecognition, and favors strategies, interests, and improvisations over the more static and homogenizing cultural tropes of rules, models, and texts” (1991: 147). Rather than a setting for actions, culture might be something people do—an *outcome* of actions. This multiplies culture radically: instead of unified “cultures,” we have an array of loosely coordinated practices that compete and collaborate, sometimes for “cultural” goals and sometimes for other goals—for instance, “technical” ones.

Annemarie Mol has advanced a radical version of this focus on practice through her ethnographic and philosophical work, which she calls “praxiography” (2002). For Mol, reality itself is not prior to practices but rather a consequence of them; in Mol’s “practical ontology” (Gad et al., 2015), actors do not act on pre-given objects, but rather bring them into being—a process she calls “enactment.” Consequently, objects acted on in many different ways become “multiples”: “more than one and less than many” (Mol, 2002: 82). A “culture,” for instance, is not one coherent thing, nor is it a set of disparate things, such that every person enacts and imagines their own in isolation.

Following Laura Devendorf and Elizabeth Goodman, I find this a useful way to approach algorithms—not as stable objects interacted with from many perspectives, but as the manifold consequences of a variety of human practices. In their study of an online dating site (2014), Devendorf and Goodman found various actors enacted the site’s algorithm differently: engineers tweaked their code to mediate between the distinctive behaviors of male and female users; some users tried to game the algorithm as they understood it, to generate more desirable matches; other users took the algorithm’s matches as oracular pronouncements, regardless of how they had been produced. No inner truth of the algorithm determined these interactions, and non-technical outsiders changed the algorithm’s function: machine learning systems changed in response to user activity, and engineers accommodated user proclivities in their code.

We can call this the *algorithms as culture* position: algorithms are not technical rocks in a cultural stream, but are rather just more water. Like other aspects of culture, algorithms are enacted by practices which do not heed a strong distinction between technical and non-technical concerns, but rather blend them together. In this view, algorithms are not singular technical objects that enter into many different cultural interactions, but are rather unstable objects, culturally enacted by the practices people use to engage with them.

This vision of algorithms as culture differs from the notion of “algorithmic culture” (Striphas, 2015), which posits algorithms as an transformative force, exogenous to culture. In this view, a movie recommender is cultural because it shapes flows of cultural material, not because its algorithmic logics are themselves cultural (Hallinan and Striphas, 2016). Nor is it what Gillespie calls “algorithms becoming culture” (2016), which happens when algorithms become objects of popular debate and targets of strategic action (e.g. fans launching a listening campaign to influence a music recommender). Rather, algorithms are cultural not because they work on things like movies or music, or because they become objects of popular concern, but because they are composed of collective human practices. Algorithms are multiple, like culture, because they *are* culture.

Methodological enactments

If we understand algorithms as enacted by the practices used to engage with them, then the stakes of our own methods change. We are not remote observers, but rather active enactors, producing algorithms as particular kinds of objects through our research. Where a computer scientist might enact algorithms as abstract procedures through mathematical analysis, an anthropologist might use ethnographic methods to enact them as rangy sociotechnical systems constituted by human practices. A computer scientist may be concerned with matters of efficiency or how an algorithm interacts with data structures; an anthropologist may care instead about how an algorithm materializes values and cultural meanings. This disparity does not mean one of us is wrong and the other right—rather, we are engaged in different projects with different goals, and just as my discipline’s methods are poorly suited to determining the efficiency of an algorithm in asymptotic time, so the computer scientist’s are poorly suited to understanding the cultural situations in which algorithms are built and implemented.

This recognition of algorithms’ multiplicity may seem destabilizing, causing trouble for efforts to hold algorithms accountable or even to identify their effects. However, I want to suggest that this approach

facilitates, rather than limits, critique, making our accounts more adequate to the practices they describe, and centering those practices as a site of dispute and potential regulation. As Marina Welker puts it in *Enacting the Corporation*: “How we construct corporations as actors has crucial entailments for how we assign responsibilities to them, and vice versa” (2014: 4). We can replace “corporations” with “algorithms” in that sentence, but we do not have to: the algorithms we care about are very often corporate products, and they seem more similar to corporations themselves—in their heterogeneity, diffuseness, and legal architecture—than to textbook algorithms like the bubble sort that I began with. Presuming that algorithms must have a stable and coherent existence makes it harder, not easier, to grapple with their production and ongoing maintenance.

Algorithmic auditing provides a useful case for examining how critical methods enact algorithmic objects. These projects, inspired by historical efforts to expose housing discrimination, treat algorithmic systems as black-box functions: hidden operations that turn inputs into outputs, like mortgage applications into home loans (Diakopoulos, 2013; Sandvig et al., 2014). By varying inputs and examining the corresponding outputs (e.g. constructing personae that vary by apparent race), audit studies can demonstrate “disparate impact”—differences in outcome that affect legally protected classes of people and thus invoke regulatory response. What they cannot do is explain conclusively how that disparate impact came about.

By treating the “inside” of the algorithm as unknowable, these approaches participate in enacting an understanding of the algorithm as a black box, as knowable only through the relation between inputs and outputs. This is not to say that audit approaches are responsible for algorithmic secrecy—they are clearly responding to other efforts to keep algorithmic operations hidden—but they are part of a set of coordinated practices through which algorithms becomes understood as, and remain, secret.

While one might presume secrecy to be a simple matter of hiding facts that could be easily revealed, secrecy in practice is not so clear; secrecy is a social process, enacted by both insiders and outsiders (this is a longstanding trope in the anthropology of secrecy; Jones, 2014; Simmel, 1906). Different methods of engaging algorithmic secrecy enact the algorithm differently, setting its boundaries in different places—at the interface, at a certain point in the code, or at the legal boundary of the corporation. If, as Frank Pasquale (2015) suggests, we take algorithmic secrecy as a legal problem, then our efforts to understand the algorithm need to involve legal reasoning. If, as Jenna Burrell (2016) suggests, one source of algorithmic opacity

is the intrinsic complexity of methods like neural networks, then we might try to engineer “explainable” systems (e.g. Aha, 2017). Thus, details of contract law may be as salient to the cultural functioning of algorithms as the ability of a neural network to “explain” its outputs.

As Daniel Neyland demonstrates through ethnographic work on an algorithmic accountability project, making algorithms accountable often means literally changing them—making them “account-able,” in ethnomedological jargon (2016). To make something account-able means giving it qualities that make it legible to groups of people in specific contexts. An accountable algorithm is thus literally different from an unaccountable one—transparency changes the practices that constitute it. For some critics this is precisely the point: the changes that transparency necessitates are changes that we want to have. This is a plain example of how different efforts to enact an object are both coordinated with each other and potentially in conflict. Transparency is not a revealing of what was always there, but a methodical reconfiguration of the social scene that changes it toward particular ends (see e.g. Ananny and Crawford, 2016; Strathern, 2000; Ziewitz, 2015).

Ethnographic tactics

This discussion recasts the anxiety described at the beginning of this article: concerns about the proper definition of “algorithm” are caught up not only in the boundary work that constitutes disciplines, but also in the methods those disciplines use. As algorithms are enacted by a wider range of methods, these enactments come into conflict with each other, and the work of coordinating among them becomes more challenging. Any method for apprehending algorithms now takes place amidst this confusion.

In this situation, I have found ethnography to be a useful method for enacting algorithms. Given its history in the study of cultural difference, it is well suited to life among plural methods—for engaging the various ways that people go about their lives, rather than trying to displace them. Not everyone needs to become an ethnographer, but ethnography as a method is distinctively appropriate to understanding how diverse methods interact. Ethnography is also good for seeing algorithms *as*, rather than *in* culture—for apprehending the everyday practices that constitute them and keep them working and changing, not just for observing how they relate to a distinct and properly “cultural” domain. Ethnographic methods help us gain purchase on topics that concern critical scholars: the interrelation of people and computers, the local production of abstract representations, and the human sensemaking that pervades algorithmic operations, locating these

within sociocultural contexts that are diverse and changing.

In the remainder of this article, I offer a set of ethnographic tactics following the tradition of the “ethnography of infrastructure” (de Certeau, 1984; Star, 1999), which has directed researcher attention to the often-neglected cultural features of sociotechnical systems. These tactics are not unique to algorithmic objects—they have their origins in a variety of ethnographic domains. Nor should we expect every new object to engender brand new tactics; indeed, one appealing feature of seeing algorithms *as*, rather than *in*, culture is a rejection of the idea that they are new or unique at all. Where many methods for critically engaging algorithms take secrecy as one of algorithms’ defining features, I try here to offer paths away from secrecy, so that ethnographers might ask and answer other questions. Given traditional ethnographic concerns with access, this is a challenge, requiring openness to fieldwork that diverges from canonical imaginaries. I have found these tactics especially useful for making algorithms ethnographically tractable. They are a pragmatic counterpart to the conceptual argument advanced thus far—a set of theoretically informed “tricks of the trade” (Becker, 1998; Star, 1999).

Scavenge

Algorithms are not the only obscure objects ethnographers have tried to study. Hugh Gusterson has studied the culture of nuclear weapons scientists—an extraordinarily secretive group (1996, 2004). Unable to access their workplaces, Gusterson developed an ethnographic method he called “polymorphous engagement”: this meant “interacting with informants across a number of dispersed sites, not just in local communities, and sometimes in virtual form; and it mean[t] collecting data eclectically from a disparate array of sources in many different ways” (1997: 116). For Gusterson, these sources included the local cafeteria, birthday parties, newspaper articles, and a variety of other heterodox “sites.”

Although this heterogeneous and apparently un disciplined approach to ethnographic data collection seems a departure from the idealized image of a fieldworker embedded long-term in a bounded society, it retains what Gusterson describes as “the pragmatic amateurism that has characterized anthropological research” since its origins (Gusterson, 1997: 116). Ethnographers have always gleaned information from diverse sources, even when our objects of study appear publicly accessible. Moreover, the scavenger replicates the partiality of ordinary conditions of knowing—everyone is figuring out their world by piecing together heterogeneous clues—but expands on them by tracing

cultural practices across multiple locations (Marcus, 1995) and through loosely connected networks (Burrell, 2009). These are “entry points, rather than sites,” as Burrell has suggested the networked ethnographer should seek (2009: 190). Access is not a precondition for all anthropological knowledge; as Ulf Hannerz writes, “ethnography is an art of the possible” (2003: 213).

A great deal of information about algorithmic systems is available to the critic who does not define her object of interest as that which is off limits or intentionally hidden. If our interest is not in the specific configuration of a particular algorithm at one moment in time, but in the more persistent cultural worlds algorithms are part of, then useful evidence is not bounded by corporate secrecy. In my own research, I learned from off-the-record chats with engineers about industry scuttlebutt, triangulated with press releases and the social media updates of my interlocutors. Sometimes, interviewees stuck resolutely to the company line; other times, often after several interviews, they spilled the beans. In academic and industry conference hallways, people working in diverse sites talked across their differences and around their various obligations to secrecy, providing a rich source of information about how algorithms and their meanings vary. On mailing lists, in patent applications, and at hackathons, I found arguments, technical visions, and pragmatic bricolage. “Algorithms” manifest across these sites differently: a conference presentation that evaluates an algorithm “properly” for its accuracy in predicting user ratings is followed by a hallway conversation about how that metric is useless in practice, or how the algorithm is too complex to be worth implementing at scale. There is much to be scavenged if we do not let ourselves be distracted by conspicuous barriers to access.

Attend to the texture of access

Nor is access as straightforward as it might seem. Achieving access remains a dream and challenge for would-be fieldworkers, granting the right to say “I was there,” but it may be more important for asserting one’s anthropological bonafides than it is for doing good ethnography—interpreting the ordinary cultural patterns and practices that make up human life, with or without algorithms. This is largely because the Malinowskian imaginary of fieldwork access, in which it happens suddenly and thoroughly—“Imagine yourself suddenly set down surrounded by all your gear, alone on a tropical beach close to a native village” (Malinowski, 1922: 4)—is just that: imaginary. In practice, “access” is a protracted, textured practice that never really ends, and no social scene becomes simply available to an ethnographer because she has shown up.

Rather, ordinary social interaction is marbled with secrecy (Jones, 2014).

I supplemented my own scavenging ethnography with a summer internship at a music recommendation company, where I was free to roam around the office and interview employees. But even in the open plan office, there were always further barriers to access. Conversations in email threads or chat rooms I wasn’t privy to, closed meetings, and coordination with companies external to the office meant that access was not a one-time achievement, but rather a continuous (and exhausting) process. Knowledge might be hidden behind non-disclosure agreements, taciturn interviewees, or inside jokes.

Although it often felt like I was being excluded as an outsider ethnographer, this situation was not unique to me. For people all over the company and the industry more broadly, everyday work was marked by varying levels of access and obscurity. Casper Bruun Jensen has described these “asymmetries of knowledge” (2010), arguing that challenges to access—hidden meetings, reluctant interlocutors, non-disclosure agreements—are part of the field, not simply barriers around it. The field is “a partially existing object emerging from multiple sites of activity that are partly visible, partly opaque to all involved actors, including the ethnographer” (Jensen, 2010: 74). Not even people on the “inside” know everything that is going on, both because algorithms can be quite complex and distributed (Seaver, 2013) and because that is how human social life works more generally. The field site is not a black box that can be simply opened.

Rather than thinking of access as a perimeter around legitimate fieldwork, the scavenging ethnographer can attend to access as a kind of texture, a resistance to knowledge that is omnipresent and not always the same. These challenges are data themselves—about the cultural life of algorithmic systems, how their secrecy is constituted in practice, what kinds of information are so important that they must be kept secret, and what kinds of information are so important that they must be widely known. Ethnographic projects, like the algorithmic systems we want to study, “are characterized by limited presence, partial information and uncertain connections” (Jensen, 2010: 74). Through paying attention to the texture of access, the ethnographer learns about how knowledge circulates, information that is practically useful but also a research outcome in its own right: an algorithm’s edges are enacted by the various efforts made to keep it secret.

Treat interviews as fieldwork

Because the scavenging ethnographer is highly mobile, her fieldwork is likely to be interview-centric; with

less time spent in any given location, it is challenging to settle into idealized participant observation (Hannerz, 2003). Multi-sited ethnographers are often anxious that this reliance on interviews renders their work less ethnographic, because interviews are commonly understood as artificial situations created by researchers. In anthropological methods talk, interviews are cast as the abject other of participant observation: they merely reflect what people *say*, not what they *do*. We all know that talkers dissimulate, misremember, or selectively highlight features of their lives. Interviews are thus framed as unreliable—primarily a source for statements to be falsified by proper fieldwork. Ironically, this framing values participant observation much like how corporations value behavioral data analytics: people *say* one thing (as indicated by their favorite artist on Facebook), but they *do* another (as indicated by their listening history on Spotify).

However, it is worth considering interviews as a form of cultural action themselves—not an artificial situation constructed by researchers, but part of the world in which research subjects live and make meaning. The people who work in and around algorithmic systems live in what Jenny Hockey calls an “interview culture”—they know what interviews are, they witness them conducted regularly in a variety of media, they’ve likely been interviewed before, and often, they’ve conducted interviews themselves (Hockey, 2002; and see Forsey, 2010; Skinner, 2012). When the researcher organizes an interview, she is setting up a known kind of interaction with its own tacit rules and meanings. Interviews do not extract people from the flow of everyday life, but are rather part of it.

Here is an partial list of interviews I conducted during fieldwork: meeting for coffee with an engineer in a San Francisco coffeeshop; setting up a Skype conversation with an interlocutor who insisted that I send him a Google Calendar invite so that our meeting would show up on his work calendar; interviewing a team of research scientists over lunch in the restaurant next door to their office; chatting in a bar with a former employee of a music streaming service that was slowly going out of business; strolling in a park with a long term informant who had become a friend; walking with an academic lab director who insisted we could only talk while he was running his on-campus errands.

None of these interactions were unusual for my interviewees. They fit me into existing patterns in their lives—often in ways that made my work difficult, giving me only 30 distracted minutes or making it hard to take notes or use my audio recorder. They treated me like a prospective hire, a supervisor, an advisee, a journalist, a friend, or a therapist. In these variously formatted conversations, algorithmic concerns manifested in many ways: as technical puzzles worked out

with colleagues over lunch, as sources of anxiety or power, as marketing tools, or even as irrelevant to the real business of a company. Treating interviews as fieldwork does not just reduce the ethnographer’s anxiety about relying on them—it broadens her attention, turning the mundane mechanics of arranging and conducting conversations into material for analysis.

Parse corporate heteroglossia

While I was in the field in January 2014, a new music streaming service named Beats Music launched. An extended commercial featured a manifesto, which lauded human musicality and criticized algorithmic processing, read over scratchy, organic animations: silhouettes kissing, turntables spinning, a sailboat tossed on stormy water made of 1s and 0s.

What if you could always have the perfect music for each moment, effortlessly? Drives would be shorter. Kisses, deeper. Inspiration would flow, memories would flood. You’d fall in love every night. [...] And to do that you’ll need more inside your skull than a circuit board. [...] We’ve created an elegant, fun solution that integrates the best technology with friendly, trustworthy humanity—that understands music is emotion, and joy, culture . . . and life.

The scavenging ethnographer will at some point encounter a document like this—an advertisement or press release or blog post from a company executive, which, in spite of claiming an author, seems certainly written by committee. Beats’ manifesto evinces what Mikhail Bakhtin called “heteroglossia,” the quality of a text that speaks with many voices at once (1982). In the short extract above, at least two voices are evident: the flowery evocation of kisses, inspiration, memories, and love and the business language of “elegant, fun solutions.” The ad was a paean to the human curators Beats claimed distinguished it from competitors, a jeremiad against algorithms from a software company, and an awkward corporate effort to integrate technology and humanity.

Corporate speech is often heteroglot, but for Bakhtin and the linguistic anthropologists who follow him, heteroglossia is not just a consequence of corporate authorship; it is an ordinary feature of language: “In the reality of everyday life [...] the speech of any one person is filled by many different *voices* or linguistically constructed personae” (Duranti, 1997: 75). Identifying heteroglossia requires long-term familiarity—without familiarity, people tend to assume that voices are necessarily singular and that variation is evidence of incoherence and contradiction rather than multiplicity (Smith, 2004).

Linguistic anthropologists have made this case for people, but for corporations its necessity is even clearer: outsiders often attribute singular agency and voice to corporations composed of hundreds or thousands of employees, working in scores of internal groups. Thus, we hear that “Facebook” has claimed or done something or that “Spotify” has a particular point of view. But while managers may try to coordinate their work, nothing intrinsically binds an engineering team to a social media intern or a company founder. Especially in young companies or those in transformation, the institutionalizing forces that work to align these various voices are weak, and obvious heteroglossia in public statements is one notable consequence.

As one of my interlocutors—an engineer with another company—put it on social media, Beat’s manifesto was “bullshit”: while it aired, Beats was advertising for algorithm engineering positions, to work on the system that would recommend its curators’ playlists to users. At a conference the previous year, I had even met one of those engineers, who bragged about the company’s technical sophistication. Speaking through advertisements, job postings, and its engineers, the company said many things at once. Across these various channels, and between the many voices within them, “algorithms” were different things: precious intellectual property, incompetent calculators, or the heralds of a new age of technical sophistication.

Heteroglossia is a resource and a hazard for the ethnographer: catching corporate messages as they move in and out of phase with each other can reveal the interplay of practices within the corporation; it is not merely evidence of corporate dissimulation. The ethnographer needs to take care to resist interpretations that cast corporations as singular actors with plain intentions expressed in public statements. Relying on only one channel, or trying to find a presumed latent coherence in it, oversimplifies corporate action. Some incoherence is to be expected.

Beware irony

If careful parsing is important for making sense of corporate speech, it is doubly important for interpreting the speech of computer programmers. As ethnographers of computing cultures have noted (e.g. Coleman, 2015), programmers are especially inclined toward irony and jokes, making the ability to parse layered meanings crucial for understanding what they say. Recall that the central purpose of Clifford Geertz’s “thick description” was to detect irony—the difference between a blink, a wink, “fake-winking, burlesque-fake-winking, rehearsed-burlesque-fake-winking” and the combinatorial elaboration of layered meanings

which subtend activities which, on the surface, may appear to be the same (1973: 6–7). Only through deep engagement and richly contextual description could the ethnographer distinguish such variety—or, in other words, be in on the joke. Superficial accounts risk taking ironic statements literally or missing the conflicted experience of programmers negotiating between different sets of values.

In my own fieldwork, I met many commercially employed programmers who were deeply ambivalent about their own work or their industry: some had previously been academics or musicians and felt like they had sold out; others felt a moral charge to pursue their own teams’ work but felt guilty about broader industry dynamics. This ambivalence often manifested in ambiguous claims or jokes about things like data, markets, or algorithms. One of my interlocutors often joked that data had “forced” him to make a design decision that was, in context, clearly a matter of personal preference. Out of context, his remarks were interpreted as evidence of what Kate Crawford has called “data fundamentalism”—“the notion that correlation always indicates causation, and that massive data sets and predictive analytics always reflect objective truth” (2013). In context, it seemed evidence more of parody or resigned irony than of enthusiastic belief.

The attribution of fundamentalisms—technological determinism, naive economism, or hyper-rationalism—to computer programmers may, in some cases, indicate more about critics’ inability to parse ironic speech than it does about technologists’ simplistic beliefs. My technical interlocutors read critical literature (I encountered the Crawford piece cited above shared by my interlocutors on social media), and they criticized thoughtless uses of algorithmic processing; but they also knew how to work in that language, either to persuade believers or to crack jokes at their expense. Describing their statements as “fundamentalism” fails to capture the complex ways that data is tactically used in practice, and it risks obscuring local critiques. As Judith Nagata has argued, “The *fundamentalist* epithet is often a form of verbal ammunition against ideological and political enemies and ‘Others,’ recalling old orientalisms, and usually imposed unilaterally, whether by academics, popular opinion, or the media” (2001: 495). Given programmers’ tendency to irony, long-term engagement is crucial for ethnographers “to understand, not only the substance and content of beliefs, but also their context, source, and application” (Nagata, 2001: 481). Algorithmic systems are not necessarily populated by data fundamentalists, but rather by diverse and ambivalent characters, working in contexts and with meanings that cannot be simply read or guessed at, but which must be

found empirically. It is too easy (and too common) to try reading programmers' motivations off of algorithmic systems and to conclude that the programmers themselves must be algorithmic—limited by naive and rigid assumptions about human life. Ironically, this mistake echoes a common humanistic critique of Big Data: that it reduces people to decontextualized formalisms. As Clifford Geertz put it 45 years ago, both of these approaches suffer from trying “to understand [people] without knowing them” (1973: 30). Pace Dourish, I would argue that avoiding such readings is a serious matter “of ethnographic responsibility and practical politics,” allowing us to grasp the social dynamics within algorithmic systems and to communicate with the people there about those systems' consequences (Dourish, 2016: 2).

Conclusion

I have argued for the merits of understanding algorithms as intrinsically cultural—as enacted by diverse practices, including those of “outside” researchers. Approaching algorithms ethnographically enacts them as part of culture, constituted not only by rational procedures, but by institutions, people, intersecting contexts, and the rough-and-ready sensemaking that obtains in ordinary cultural life. This enactment differs from canonical expert enactments, which hold algorithms to be essentially abstract procedures, and it differs from other cultural approaches to algorithms that try to locate them as forces on culture's boundaries. Ethnography helps us to attend to empirical situations which are not necessarily stable or coherent.

Ethnography provides a useful orientation for entering and understanding worlds of meaning-laden practice, but conventional understandings of algorithms as defined by secret procedure suggest that ethnographic approaches are infeasible without a level of access that cannot realistically be obtained. The tactics I have laid out here are techniques for routing around that challenge; they work to enact algorithms not as inaccessible black boxes, but as heterogeneous and diffuse socio-technical systems, with entanglements beyond the boundaries of proprietary software.

While I insist on imagining alternatives to visions of algorithms as essentially secret, this style of ethnographic enactment does not answer all the questions that people might want to ask of them. Questions about the particular workings of particular algorithms at particular moments in time remain broadly unanswerable so long as corporations are able to hide behind legal and technical secrecy. Nonetheless, a sense of the algorithm as multiple, and of ethnography as a practice for producing and participating in plural enactments of algorithms, offers fruitful avenues for

producing actionable knowledge in spite of such secrecy.

Acknowledgments

I am grateful to the many colleagues and audiences who provided advice, feedback, and support as I developed the argument in this article, including: Bill Maurer, Paul Dourish, Malte Ziewitz, Stuart Geiger, Guilherme Sanches de Oliveira, Rachel Bergman, Tarleton Gillespie, Mary Gray, Nancy Baym, Lana Swartz, Kevin Driscoll, Sarah Brayne, Andrea Alarc en Sojet, Moon Duchin, Riccardo Strobino, the editors of this Special Issue, and the anonymous peer reviewers. I would also like to thank my anonymous ethnographic interlocutors.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was funded in part by National Science Foundation (Doctoral Dissertation Research Improvement Grant 1323834), the Wenner-Gren Foundation (Dissertation Fieldwork Grant 8797), and the Intel Science and Technology Center for Social Computing at the University of California, Irvine. I am also grateful to the Department of Anthropology at the University of California, Irvine for seed research funding, to the Algorithms in Culture project at the University of California, Berkeley for funding to attend the workshop where this article was developed, and to the Social Media Collective at Microsoft Research, New England for financial support during the first drafting of this argument.

Notes

1. This is a paraphrase from memory.
2. See the reading list compiled by Gillespie and Seaver (2015) here: <https://socialmediacollective.org/reading-lists/critical-algorithm-studies/>
3. Repeatedly step through a list, comparing adjacent items and swapping them if they are out of order, until nothing remains to be swapped.

References

- Abu-Lughod L (1991) Writing against culture. In: Fox RG (ed.) *Recapturing Anthropology*. Santa Fe, New Mexico: School of American Research Press, pp. 137–162.
- Aha DW (2017) IJCAI 2017 Workshop on Explainable Artificial Intelligence (XAI). Available at: <http://home.earthlink.net/~dwaha/research/meetings/ijcai17-xai/> (accessed 10 October 2017).
- Ananny M and Crawford K (2016) Seeing without Knowing: Limitations of the Transparency Ideal and Its Application to Algorithmic Accountability. *New Media & Society*. Online before print, December.

- Bakhtin M (1982) In: Holquist M (ed.) *The Dialogic Imagination: Four Essays* (Transl. Caryl Emerson). Austin: University of Texas Press.
- Barocas S, Hood S and Ziewitz M (2013) Governing algorithms: A provocation piece. Available at: <http://governalgorithms.org/resources/provocation-piece/> (accessed 10 October 2017).
- Becker HS (1998) *Tricks of the trade: How to think about your research while you're doing it*. Chicago: University of Chicago Press.
- Bourdieu P (1972) *Outline of a Theory of Practice*. Cambridge: Cambridge University Press.
- Burrell J (2009) The field site as a network: A strategy for locating ethnographic research. *Field Methods* 21(2): 181–199.
- Burrell J (2016) How the Machine ‘Thinks:’ Understanding Opacity in Machine Learning Algorithms. *Big Data & Society* 3(1).
- Coleman G (2015) The anthropological trickster. *HAU: Journal of Ethnographic Theory* 5(2): 399–407.
- Crawford K (2013) The hidden biases in Big Data. *Harvard Business Review*. 1 April. Available at: <https://hbr.org/2013/04/the-hidden-biases-in-big-data> (accessed 10 October 2017).
- de Certeau M (1984) *The Practice of Everyday Life*. Berkeley: University of California Press.
- Devendorf L and Goodman E (2014) *The Algorithm Multiple, the Algorithm Material*. Contours of Algorithmic Life, UC Davis, May.
- Diakopoulos N (2013) Algorithmic Accountability Reporting: On the Investigation of Black Boxes. *A Tow/Knight Brief*. New York: Columbia Journalism School, Tow Center for Digital Journalism.
- Dourish P (2016) Algorithms and their others: Algorithmic culture in context. *Big Data & Society* 3(2): 1–11.
- Duranti A (1997) *Linguistic Anthropology*. Cambridge, UK: Cambridge University Press.
- Forsey MG (2010) Ethnography as participant listening. *Ethnography* 11(4): 558–572.
- Fuller S (1991) Disciplinary boundaries and the rhetoric of the social sciences. *Poetics Today* 12(2): 301–325.
- Gad C, Jensen CB and Winthereik BR (2015) Practical Ontology: Worlds in STS and Anthropology. *NatureCulture* 3: 67–86.
- Gal S and Irvine J (1995) The boundaries of languages and disciplines: How ideologies construct difference. *Social Research* 62(4): 967–1001.
- Geertz C (1973) *The Interpretation of Cultures*. New York, NY: Basic Books.
- Gieryn T (1983) Boundary-work and the demarcation of science from non-science: Strains and interests in professional interests of scientists. *American Sociological Review* 48: 781–795.
- Gillespie T (2016) Algorithm. In: Peters B (ed.) *Digital Keywords: A Vocabulary of Information Society and Culture*. Princeton, NJ: Princeton University Press.
- Gillespie T and Seaver N (2015) Critical algorithm studies: A reading list. Available at: <https://socialmediacollective.org/reading-lists/critical-algorithm-studies/> (accessed 10 October 2017).
- Gusterson H (1996) *Nuclear Rites*. Berkeley: University of California Press.
- Gusterson H (1997) Studying up revisited. *PoLAR: Political and Legal Anthropology* 20(1): 114–119.
- Gusterson H (2004) *People of the Bomb*. Minneapolis: University of Minnesota Press.
- Hallinan B and Striplas T (2014) Recommended for You: The Netflix Prize and the Production of Algorithmic Culture. *New Media & Society* 18(1): 117–137.
- Hannerz U (2003) Being there... and there... and there! *Ethnography* 4(2): 201–216.
- Helmreich S (2001) After culture: Reflections on the apparition of anthropology in artificial life, a science of simulation. *Cultural Anthropology* 16(4): 612–627.
- Hockey J (2002) Interviews as ethnography? Disembodied social interaction in Britain. In: Rapport N (ed.) *British Subjects: An Anthropology of Britain*. Oxford: Berg, pp. 209–222.
- Jensen CB (2010) Asymmetries of knowledge: Mediated ethnography and ICT for development. *Methodological Innovations Online* 5(1): 72–85.
- Jones GM (2014) Secrecy. *Annual Review of Anthropology* 43(1): 53–69.
- Malinowski B (1922) *Argonauts of the Western Pacific*. London: Routledge & Kegan Paul, Ltd.
- Marcus G (1995) Ethnography in/of the World System: the Emergence of Multi-Sited Ethnography. *Annual Review of Anthropology* 24: 95–117.
- Mol A (2002) *The Body Multiple: Ontology in Medical Practice*. Durham, NC: Duke University Press.
- Nader L (1969) Up the anthropologist: Perspectives gained from studying up. In: Hymes D (ed.) *Reinventing Anthropology*. New York, NY: Vintage Books.
- Nagata J (2001) Beyond theology: Toward an anthropology of ‘Fundamentalism.’ *American Anthropologist* 103(2): 481–498.
- Neyland D (2016) Bearing account-able witness to the ethical algorithmic system. *Science, Technology & Human Values* 41(1): 50–76.
- Ortner S (1984) Theory in anthropology since the sixties. *Comparative Studies in Society and History* 26(1): 126–166.
- Pasquale F (2015) *The Black Box Society: The Secret Algorithms that Control Money and Information*. Cambridge, MA: Harvard University Press.
- Sandvig C, Kevin H, Karrie K, et al. (2014) An Algorithm Audit. In: Seeta Peña Gangadharan (ed.) *Data and Discrimination: Collected Essays*. Washington, DC: New America Foundation, pp. 6–10.
- Seaver N (2013) Knowing algorithms. Paper presented at Media in Transition 8, Cambridge, MA. Available at: <http://nickseaver.net/papers/seaverMiT8.pdf> (accessed 10 October 2017).
- Skinner J (ed.) (2012) *The Interview: An Ethnographic Approach*. London: Bloomsbury Academic.
- Simmel G (1906) The Sociology of Secrecy and of Secret Societies. *The American Journal of Sociology* 11(4): 441–498.
- Smith AL (2004) Heteroglossia, ‘common sense,’ and social memory. *American Ethnologist* 31(2): 251–269.

- Star SL (1999) The Ethnography of Infrastructure. *American Behavioral Scientist* 43(3): 377–391.
- Strathern M (1995) The nice thing about culture is that everyone has it. In: Strathern M (ed.) *Shifting Contexts: Transformations in Anthropological Knowledge*. London: Routledge.
- Strathern M (2000) *Audit Cultures: Anthropological Studies in Accountability, Ethics and the Academy*. New York: Routledge.
- Striphas T (2015) Algorithmic culture. *European Journal of Cultural Studies* 18(4–5): 395–412.
- Titlow JP (2013) Screw algorithms! The new music service from beats uses celebrity curators to show you new music. *Fast Company*. Available at: <https://www.fastcompany.com/3019830/screw-algorithms-the-new-music-service-from-beats-uses-celebrity-curators-to-show-you-new-mu> (accessed 10 October 2017).
- Totaro P and Domenico N (2014) The Concept of Algorithm as an Interpretative Key of Modern Rationality. *Theory, Culture & Society* 31(4): 29–49.
- Welker M (2014) *Enacting the Corporation: An American Mining Firm in Post-Authoritarian Indonesia*. Berkeley: University of California Press.
- Ziewitz M (2015) Governing algorithms: Myth, mess, and methods. *Science, Technology & Human Values* 41(4): 3–16.