"Be careful; things can be worse than they appear": Understanding Biased Algorithms and Users' Behavior around Them in Rating Platforms

Motahhare Eslami, Kristen Vaccaro, Karrie Karahalios[†], Kevin Hamilton
University of Illinois at Urbana-Champaign, [†]Adobe Research
{eslamim2, kvaccaro, kkarahal, kham}@illinois.edu, [†]karrie@adobe.com

Abstract

Awareness of bias in algorithms is growing among scholars and users of algorithmic systems. But what can we observe about how users discover and behave around such biases? We used a cross-platform audit technique that analyzed online ratings of 803 hotels across three hotel rating platforms and found that one site's algorithmic rating system biased ratings, particularly low-to-medium quality hotels, significantly higher than others (up to 37%). Analyzing reviews of 162 users who independently discovered this bias, we seek to understand if, how, and in what ways users perceive and manage this bias. Users changed the typical ways they used a review on a hotel rating platform to instead discuss the rating system itself and raise other users' awareness of the rating bias. This raising of awareness included practices like efforts to reverse-engineer the rating algorithm, efforts to correct the bias, and demonstrations of broken trust. We conclude with a discussion of how such behavior patterns might inform design approaches that anticipate unexpected bias and provide reliable means for meaningful bias discovery and response.

Keywords: Algorithm Audits; Algorithm Awareness; Algorithm Bias; Rating Platforms; Situated Action

Introduction

Algorithms are powerful. They collect, process, and present information in today's online world; and in doing so, they exert influence over users' interaction with the system. Rating algorithms are one example that their outputs — i.e., business ratings based on online reviews — significantly impact users' behavior and accordingly the success of a business. A simple half-star improvement on a Yelp rating, for instance, results in a 30-49% higher likelihood of selling out the seats for a restaurant (Anderson and Magruder 2012).

While rating algorithms are influential, little about how they work is made public. These algorithms' internal processes, and sometimes their inputs, are usually housed in black boxes, both to protect intellectual property and to prevent reviewers from gaming business ratings. Computing a raw average of users' reviews is a simple way to calculate a business rating; however, many rating platforms do not use this approach. Amazon, for instance, calculates a product's overall rating by taking into account factors including the age of the review, helpfulness votes and whether the reviews are from verified purchases (Bishop 2015). Some other platforms, like Yelp (Yelp 2010), calculate a business rating by computing a raw average of customer reviews, but only of reviews that their rating algorithms classify as authentic or "not fake." While we understand the overview of these algorithms, their details are proprietary.

The power and opaqueness of algorithmic rating systems have raised concerns about the bias they might introduce into online ratings. As an example, in May 2016, Australian Uber drivers accused the company of slowly decreasing their ratings to suspend them and then charge higher commissions to be reinstated. The president of the Ride Share Drivers' Association of Australia noted that "*the lack of transparency makes it entirely possible for Uber to manipulate the ratings*" (Tucker 2016). Other algorithmic rating systems such as Yelp (Fowler 2011) and Fandago (Hickey 2015) have faced similar criticisms, as have other algorithmic systems including search engines (Hannak et al. 2013; Kliman-Silver et al. 2015; Kulshrestha et al. 2017), online advertising (Sweeney 2013; Datta, Tschantz, and Datta 2015) and e-commerce websites (Hannak et al. 2014). These issues have given rise to a growing area of research, designing algorithms audits, that aims to detect algorithmic systems' potential biases (Sandvig et al. 2014).

Detecting the existence of bias in an algorithmic system, however, is not sufficient to understand its impact on users. An important factor in discovering the effects of an unjust or biased algorithm is understanding users' awareness of and behavior around it. Recent studies have investigated users' awareness of and interaction with algorithms — including news feed curation algorithms (Eslami et al. 2015; 2016; Rader and Gray 2015) and ridesharing management algorithms (Lee et al. 2015) — but how users perceive and manage the bias that an algorithm brings to their online experience is an open question.

In this paper, we seek to fill these gaps by investigating algorithmic bias and users' awareness of and behavior around the bias in hotel rating platforms. An initial study suggested that a potential bias on a hotel rating platform (Booking.com) skewed low review scores upwards. To analyze this potential bias, we used a cross-platform audit technique comparing the outputs of Booking.com and two other popular hotel rating platforms. Analyzing the ratings of 803

hotels showed that Booking.com's rating system biased ratings of hotels, particularly low-to-medium quality hotels, to be significantly higher than other platforms (up to 37%).

We employed a mixed-method design to study users' behavior around this bias. First, we applied a computational technique to identify the users who noticed the bias; next, we conducted qualitative analysis over their reviews to understand how users behaved around the bias. We found 162 users who independently discovered the algorithm's bias through their regular use. These users, rather than contributing the usual review content (i.e., informing other users about their hotel stay experience), adopted an "auditing" practice. When confronted by a higher than intended review score, they used their review to raise the bias awareness of other users on the site. To do so, they wrote about how they: engaged in activities such as trying to manipulate the algorithm's inputs to look into its black-box, tried to correct the bias manually, and illustrated a breakdown of trust.

While the work we present here focuses on rating algorithms, we are more broadly concerned with the role of algorithmic bias and users' behavior around it in non-transparent algorithmic systems in general. We conclude with connections to both theory and design. We discuss how bias awareness can shift users' attention from their own experience to the system as a whole; this suggests the possibility of users collaborating to bring an algorithmic bias to the surface via a "collective audit." We also explore design approaches that bring *actionable* transparency to algorithmic systems, building a more trustworthy and engaging interaction between users and the system.

Algorithms And Bias

Algorithms help us to interact with large amounts of data by automating the curation of online content (Lustig et al. 2016). They are not, however, infallible; they might introduce bias into a system due to their probabilistic nature, imperfect human logic used in their development, or non-representative input or training data from biased individuals (Seaver 2013; Diakopoulos 2014). Such biases, regardless of their source, might result in unintended consequences with negative impacts, like an image tagging algorithm labelling images of Black people with tags such as "ape" (Hern 2015), and arguments over whether the Chicago's police crime prediction algorithm is racist (Stroud 2014).

Auditing Algorithms: Techniques and Challenges

The presence of biases in algorithmic platforms has opened up new approaches for interrogating algorithms. In "algorithm audits" researchers employ different techniques inspired by traditional audit studies (such as field experiments to detect realtors' racial discrimination in choosing which homes to present to clients) to understand biases or illegal behaviors an algorithm might introduce to a system (Sandvig et al. 2014). The potential risks of biased algorithms have also prompted governments to call for regulation of algorithmic systems in an attempt to increase algorithmic transparency and prevent discrimination via algorithms (Goodman and Flaxman 2016; WhiteHouse 2016). The black box nature of algorithms, however, makes auditing them difficult. Therefore, many algorithm audits are designed to detect algorithm bias "from the outside." Here, we categorize these techniques, depending on whether an audit is performed within one platform or across two or more platforms.

Within-Platform Audits In within-platform audit techniques, researchers issue queries as inputs to an algorithmic system and analyze the outputs. For example, Sweeney searched more than 2000 racially-associated personal names online and found that ads suggesting arrest were 25% more likely to appear for Black-identifying names (Sweeney 2013). Researchers also use scraping to audit, writing a script to conduct such processes automatically and at scale (Sandvig et al. 2014). Investigating the potential biases of online maps in representing international borders (Soeller et al. 2016) and exploring the sources of bias in social media search are some examples.

Generating or collecting inputs to feed an algorithmic system is challenging, however, particularly if the input of an algorithm is a real user's profile. To overcome this challenge, some studies use a "sock puppet" technique (Sandvig et al. 2014), in which researchers create fake users' profiles as inputs to an algorithm to analyze its outputs and investigate bias. Examples of this type of audit include creating browser agents as simulated users to detect gender bias in online advertising (Datta, Tschantz, and Datta 2015), understanding the Uber surge pricing algorithm by emulating Uber accounts (Chen, Mislove, and Wilson 2015), and discovering racial discrimination against Black users on Airbnb via creating multiple accounts (Edelman, Luca, and Svirsky 2015). Some have used this technique to understand which specific input data in a real user's profile maps to outputs (e.g., ads and recommendations) by associating shadow and real user accounts and comparing the outputs (Lécuyer et al. 2014).

Scraping and sock puppet audit techniques may be difficult or even illegal to use under a platform's terms of service (ToS). As ToS are included in the Computer Fraud and Abuse Act (CFAA), violations may be labeled a federal crime. Therefore, some researchers recruit real users (similar to traditional audits) to use an algorithmic system to analyze the outputs and detect potential biases. This method, the "crowdsourced audit" (Sandvig et al. 2014), has been used by researchers to examine personalization algorithms in Google search (Hannak et al. 2013; Kliman-Silver et al. 2015) and e-commerce websites (Hannak et al. 2014). While effective, it is difficult to implement at scale.

Cross-Platform Audits The main goal of within-platform audits is to see if an algorithm behaves differently across *some* categories of inputs when it should not (e.g., showing more arrest records for Black people or higher prices for users in poor neighborhoods). But what if an algorithmic system biases *all* its inputs? Detecting such biases is difficult with a within-platform audit. Auditing across platforms allows the detection of bias that skews all inputs, by comparing outputs of an algorithmic system with the outputs of other systems that have a similar intent.

However, while previous work compared different algorithmic systems' outputs around the same input, the goal

was not to audit the algorithms. For example, Muddiman compared search engines' results for the 2008 U.S. presidential candidates to gain a holistic insight into how search engines organize political information during campaigns, but not to detect potential biases in these algorithmic systems. (Muddiman 2013)). Recently, however, some journalists have started to use cross-platform audit techniques to detect bias in algorithmic platforms. An example is finding that Fandango's rating algorithm skewed their movie ratings upwards in comparison to other movie rating platforms (Hickey 2015). Inspired by such journalistic investigations, our work uses a cross-platform audit to investigate potential bias in online hotel rating systems.

Users' Behavior around (Biased) Algorithms

Detecting an algorithm's bias is the first step toward understanding its impacts. However, detection alone is not enough. Understanding whether users are aware of algorithmic biases and how they perform around such biases is important for perceiving the possible effects of algorithmic bias not only on the platform but also on users' experiences.

In some cases, the significance of an algorithm's bias can be truly understood by analyzing users' behavior around it. For example, the "Up Next" YouTube algorithm that recommends additional videos appears benign at first glance. However, an incident catalyzed by the "Reply Girls", highlighted how it could be biased. A group of users, the "Reply Girls," speculated that this algorithm prioritized videos labeled replies to a video in the "Up Next" list. By uploading sexually suggestive videos as replies to popular videos, they increased their view number and, as a result, their ad sharing revenue. Many YouTube users began campaigning against the "Reply Girls" via Youtube channels, and YouTube modified its algorithm to resolve this issue (O'Neill 2012). This algorithmic bias, however, may not have been addressed if the audience did not detect it and react against it.

Recent studies have discussed the importance of understanding users' awareness of and their behavior around algorithms (Hamilton et al. 2014; Barocas, Hood, and Ziewitz 2013). Analyzing human workers' interaction with algorithmic management in ridesharing services Uber and Lyft (Lee et al. 2015), users' awareness and their understanding of Facebook's News Feed curation algorithm (Eslami et al. 2015; Rader and Gray 2015) and the practices they use to make sense of it (Eslami et al. 2016) are some examples. However, there is still little understanding about users' behavior around biased or misleading algorithms. We seek to improve this understanding by analyzing how users understand and manage the bias of an algorithmic rating system.

Case Study: Rating Algorithms

Rating systems are one type of algorithmic system that have been suspected of being biased due to their black-box nature and their influence over business ratings. Some users, particularly business owners, argue that hidden rating algorithms might misrepresent ratings in favor of or against some businesses or products (Fowler 2011; Hickey 2015).

Booking.com is one rating platform that our initial study suggested is biased in its rating algorithm. When rating a hotel on Booking.com, users cannot provide an overall score, but instead are asked to rate different criteria such as location and staff. The rating algorithm then calculates an aggregate review score for the user (Booking.com 2015). However, while Booking.com's overall review interface suggests a lowest possible score of 1, the lowest possible output of the scoring algorithm is a 2.5 (Figure 1). That is, even if a user rates all the criteria of a hotel at the lowest value, the aggregate rating returned by the algorithm is a 2.5. To understand how much bias this discrepancy introduces to hotels' overall ratings and how users behave around it, we ask the following research questions:

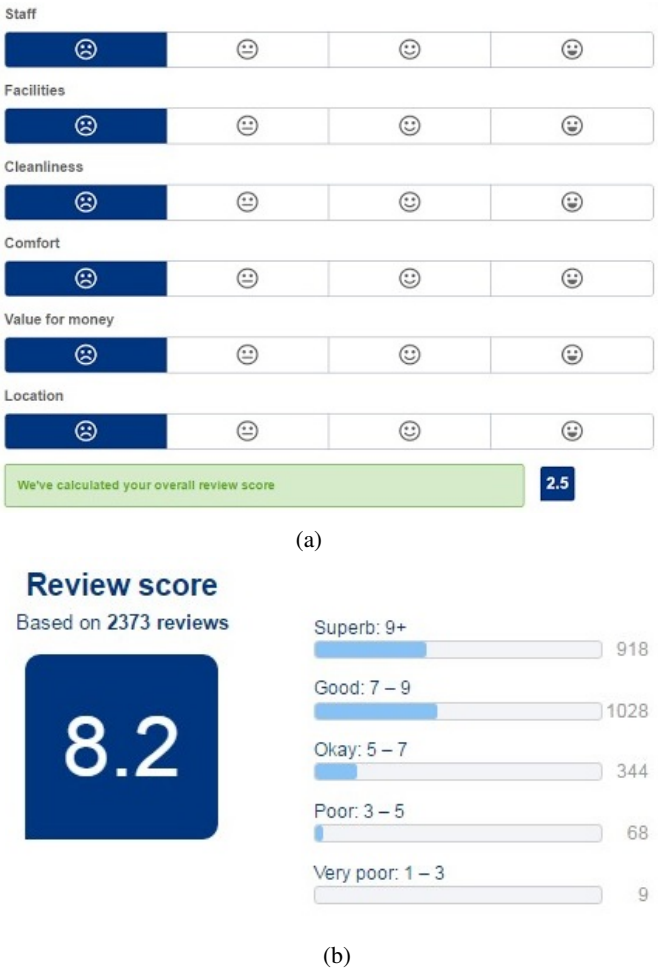

Figure 1: (a) Booking.com's user rating entry interface. With the lowest possible rating for each criterion, the resulting aggregate rating is 2.5. (b) The distribution of ratings. The lowest bin suggests the lowest rating is a value of 1.

RQ1: How much bias does Booking.com's algorithmic rating system introduce to businesses' ratings?

RQ2: Are users aware of the bias of Booking.com's algorithmic rating system? If so, how?

RQ3: How do users perceive and manage the bias that Booking.com's algorithmic rating system brings to their experience?

Table 1: Analyzing the difference of ratings between Booking.com and other rating platforms via one-sample t-test

Rating	Ratings Difference%	t-value	p-value	Effect size	M	SD
<7	Booking vs Expedia	16.26	<0.0001	1.4	27.23	19.46
	Booking vs Hotels	21.12	<0.0001	1.81	37.07	20.39
≥ 7	Booking vs Expedia	16.26	<0.0001	0.62	4.11	6.53
	Booking vs Hotels	19.69	<0.0001	0.76	5.67	7.45

Bias Detection and Quantification (RQ1)

To understand whether and how the inflation of the lowest possible review score on Booking.com introduces bias to overall hotel ratings[1], we employed a cross-platform audit technique. Since inflating the minimum scores might be a result of an inflation of all inputs, using a cross-platform rather than within-platform audit is necessary (as there would otherwise be no ground-truth for comparison). A cross-platform audit requires specific hotel's reviews and ratings from different platforms. Many websites aggregate hotel deals from travel websites; such sites, which refer to themselves as hotel metasearch engines, search several hotel booking websites to help users compare hotels across them. One metasearch engine, HotelsCombined.com, provides direct links to hotels' profiles, including their reviews and ratings from different hotel rating platforms. This site served as the data source for our study.

We collected hotel information from HotelsCombined.com for every hotel in ten random cities in the U.S.[2], leading to an initial set of 1576 hotels. We excluded any hotels that had not been rated on Booking.com and two other hotel rating platforms— Expedia.com and Hotels.com — which are among the top 10 most-used hotel booking websites[3]. The final corpus contained 803 hotels that were rated on all three hotel rating platforms. As the scales of rating platforms differ, we mapped all values to the same range (maintaining the ratio) for comparison.

Augmented Hotel Ratings

An ANOVA analysis showed a statistically significant difference between hotel ratings in the three hotel rating platforms ($F(2, 2406) = 39.9, p < 0.0001$). To discover which rating platforms were significantly different, we used a Tukey post hoc test. It revealed that Booking.com's hotel ratings were significantly different from ratings of both Expedia.com ($p < 0.01$) and Hotels.com ($p < 0.01$). There was, however, no statistically significant difference between the ratings of Expedia.com and Hotels.com ($p = 0.07$).

Figure 2 shows the ratings on Booking.com and the other platforms. Hotels with a rating lower than 7 (low-to-medium quality hotels that Booking.com considers below "good") show larger differences across platforms. To better understand this difference, we ran separate analyses on the lower

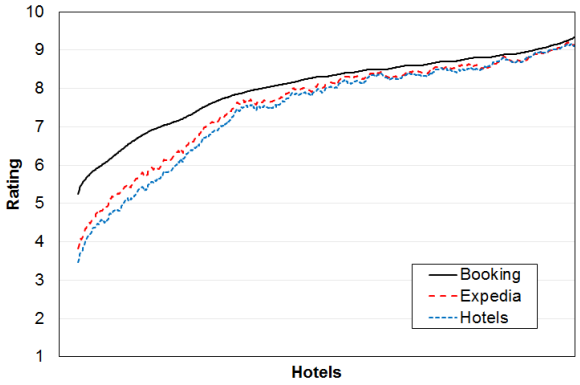

Figure 2: A comparison of hotel ratings between Booking.com and other hotel rating platforms.

and higher quality hotels. Higher quality hotels were those with ratings above 7; lower quality hotels were all remaining hotels. For each pair of significantly different platforms (Boooking.com and Expedia.com, Booking.com and Hotels.com), we measured the percentage difference of each hotel's rating. For example, if a hotel was rated 4.5 on Booking and 4 on Expedia.com, the difference measure would be 12.5%. A one-sample t-test on each group found that the difference between Booking.com and each of the other platforms is significant, with a mean difference (M) of up to 37% for lower quality hotels and around 5% for higher quality hotels, with large effect sizes (Table 1). These results indicate that Booking.com's rating system biased ratings of hotels higher than the other platforms in general. In doing so, Booking.com benefits hotels, particularly low-to-medium quality hotels.

We note that a cross-platform audit technique cannot distinguish whether a bias in the outputs of an algorithmic system arises from its inputs, the algorithm itself, or both. For example, Booking.com, Expedia.com, and Hotels.com each have their own audience, interface design (e.g., 5- vs 10-star rating scales) which can impact the inputs to their rating algorithms. Given these inherent differences between rating systems, we cannot claim the bias in Booking.com rating is only a result of its algorithm. We term this bias an "algorithmic system bias" rather than an "algorithm bias" because other elements of the Booking.com algorithmic rating system such as its inputs and users might play a role in creating this bias.

However, even without identifying the source of this bias, detecting it is important because of its impact. On aggregator systems such as HotelsCombined.com, users see different rating systems' outputs on the same page. They may assume that a 5.0 on one site is comparable to a 5.0 on another because of this presentation. But they do not see the original inputs or algorithms used in the respective sites. So even though we cannot determine whether it is Booking.com's rating algorithm alone or also input differences (e.g., the different users across each platform, different timestamps for the reviews) causing the bias, knowing that a bias exists may serve as a signal to users and help inform their interactions.

[1]Booking.com calls the scores of users' reviews a "review score" and the score of a hotel a "rating."

[2]The cities were chosen using the "randomlists" website, https://www.randomlists.com/random-us-cities

[3]http://www.toptenreviews.com/services/home/best-hotel-booking-services/

Bias Awareness (RQ2)

Rating platforms are designed for users to rate a third-party business; for example, Google's app store allows users to write reviews for applications to provide feedback about an app's issues. Researchers have used these types of reviews to detect usability issues in a system (Keertipati, Savarimuthu, and Licorish 2016). In this vein, we investigate whether users in a biased system use reviews intended for third-parties to review the system itself; i.e. whether Booking.com's users use reviews that are supposed to be about hotels to instead review Booking.com itself.

To understand whether there were users who were aware of Booking.com's rating bias and who mentioned it in their reviews, we studied user reviews from the hotels collected in the previous step. One possible method for becoming aware of a bias was to note the unintended aggregate 2.5 score when a 1 was expected. We, therefore, focused on reviews with a score of 2.5. Of 100 random reviews with a 2.5 score, eight articulated the bias to the community in their review; the reviewers described their negative hotel stay experience along with an explanation for why they suspected their review score was not the lowest possible score: "*Disgusting, roaches, uncomfortable beds[;] review needs to be less then[sic] 2.5 stars but [the system] won't let me*" (R66).

This finding inspired an expansion to study more reviews, including those with higher scores. However, the list of every Booking.com review ever written is quite long. To direct our investigation, we developed a method inspired by bootstrapping methods that start from one seed and then expand.We first extracted the three most common keywords ("2.5", "lowest", and "score") found in our initial set of 2.5 rated reviews. We then searched for these keywords altogether using Google search (specifying a Booking.com search domain). We devised a coding scheme to describe which reviewers discovered the bias. With our code, a reviewer was labeled as aware of the bias if they articulated *i*) a discrepancy between their intended review score and what the system calculated and/or *ii*) the observation that 2.5 is the lowest possible score on Booking.com.

Manually reading the top 200 search results, we identified reviewers with reviews that our code labeled as aware. We added these reviews to our set of reviews and repeated this process, extracting the three most common keywords from the new review set, searching for reviews that contained these keywords on Booking.com and coding the results. After ten iterations and over 2000 reviews, we found 162 reviewers who discussed the rating bias in their reviews. But how did these reviewers become aware of the bias?

Becoming Aware

For all of the 162 reviewers who discussed a bias in their reviews, a review score that did not match their stay experience was a prompt to discover the bias. For example, their reviews contained statements such as "*[the] overall score DOES NOT accurately reflect my opinion of this hotel*" (R96). Reviewers usually had a roughly pre-determined score in their mind to assign to their hotel stay.

Reviewers with a score of 2.5 (n=111) wrote that they had aimed for the lowest possible score for a hotel as they "*liked absolutely nothing about this accommodation*" (R5). They assigned the minimum subscore to each evaluation criteria and expected a standard minimum final score (1 or 0) that never appeared. This mismatch led them to suggest the rating bias: "*I rated this motel at 0, but the review program enters 2.5 as the low limit, so be careful; things can be worse than they appear*" (R98).

Reviewers with higher scores than 2.5 (n=50), up to 8.8, also found the calculated score did not match their desired score: "*Although the above rating indicates a 3.8, I would personally rate it about a 1.5* " (R55). Our initial hypothesis was that these reviewers had previously written 2.5-score reviews while expecting a lower score, discovered the bias, and reflected their finding in their next review(s). However, more than half of these users had only written one review with a score higher than 2.5 and still discovered the bias. These reviewers realized that "*the way [their] score is calculated doesn't give a good idea of the overall notation*" (R93).

Behaving around the Bias (RQ3)

To understand users' behavior around the bias once they were aware of it, we analyzed users' reviews using an inductive, iterative process via line-by-line open coding. First, we read all the reviews several times, before labeling each review with preliminary codes to help organize our initial insights. These codes included a set of primary themes such as announcing a mismatched review score, manipulating the algorithm, or asking for a change in the rating system.

Second, we analyzed the codes themselves to find similarities, grouping them into categories based on common properties. We arrived at three main themes. The first theme revolves around users' attempts to understand how the algorithm works. The second focuses on users' efforts to correct the bias. The third addresses the breakdown of trust between users and the system as a whole due to the bias.

Last, we conducted axial coding, analyzing the interrelationships among the themes that we had identified. We found a common theme in all three: *raising awareness through situated actions*. Users changed their usual practices around the reviews — instead of only informing other users about their hotel stay experience, they sought to raise other users' awareness of the rating bias. This change of action resonates with Suchman's idea of "situated action" in which users adjust their behavior depending on what is actually happening in an interaction with a system in a specific situation (Suchman 1987). We describe the three main themes and how the practice of raising awareness is embedded in each below.

Looking into the Black Box

> *The algorithm by Booking.com seems to be biased in the high direction.* (R55)

Reviewers became confused when confronted by a higher than intended review score. They were "*not sure how booking.com came up with the score*" (R114), expressing that the "*rating calculated by this site is confusing based on [their] feedback*" (R148). While the uncertainty resulting from mismatched review scores confused some reviewers,

it also prompted some (n=67) to further reflect on the way Booking.com calculated their review score.

Users tried to make inferences about the review ratings, discovering that "*some algorithm they have on this site is giving them [hotels] a rating better than 0*" (R54). They highlighted the calculated nature of the ratings, stating that their review score "*was automatically calculated by booking.com*" (R146) and was not given by themselves directly: "*This score is automatically calculated and even with the most unhappy face gives it a 2.5*" (R160).

To understand how the rating algorithm calculated their review scores, users attempted to prod the algorithm by entering different subscores for each evaluation criteria — the inputs of the rating algorithm:

> *I looked at what score your "algorithm" comes up [with] if I give the lowest grade on all fronts and it still comes out with a 2.5 instead of zero! So I guess your range is from 2.5 to 10, instead of 0 to 10. Nice!* (R38)

Through these experiments with the algorithm inputs, some hypothesized that the algorithm not only inflates the lowest review scores to 2.5, but also skews higher scores upwards: *Booking.com's lowest score possible is 2.5, not 0. (So while it says 5/10, we're really aiming at a 2.5/10)* (R27). These explanations match our results in RQ1, where we found that Booking.com biases the businesses ratings higher in general when compared to other hotel rating platforms.

In "The Relevance of Algorithms," Gillespie calls such user practices to reverse-engineer an algorithm "*backstage access*" (Gillespie 2012). Although algorithms in sociotechnical systems are usually hidden in black boxes, users sometimes try to make sense of them. They attempt to prod an algorithm by changing its inputs and affect its outputs for their own benefit. For example, teenagers add product names to their Facebook stories in the hope of getting more visibility from the News Feed curation algorithm (boyd 2014).

Our study, however, points to a more complex story: users might attempt to prod an algorithm and understand how it works for reasons that are *beyond* their own benefit. Confronted by a bias in their review score, reviewers tried to understand how the rating algorithm works. They, however, did not do this solely for personal knowledge; they aimed to to make others aware of the bias as well.

From Uncertainty to Awareness Raising After probing how the rating algorithm worked, many reviewers used their review as an opportunity to share their understanding of the algorithm with others. Not only did they seek to share their knowledge about how the algorithm works, they wanted to warn other users about the existing rating bias on Booking.com, and help them make more informed decisions when looking for a hotel. Therefore, reviewers began by asking others to "*take note that 2.5 is the absolute minimum you can score a hotel*" (R15). They then used their inferences about the algorithm to suggest that other users not only "*ignore the 2.5 score*" (R20) but also "*ignore the ratings*" (R109) altogether. Telling readers to "*be forewarned*" (R58) was a common theme via warnings like "*Pay no attention to the score: the worst is 2.5 thus scores don't help*" (R25) or "*Don't be fooled by the ratings*" (R27). As a result, reviewers urged other users to refer to the text of the reviews rather than their ratings to learn about the reviewers' true stay experience:

> *PLEASE READ THESE COMMENTS - I AM MAKING IT IN THE INTEREST OF FUTURE GUESTS. The rating as calculated by Booking.com does not do justice to our experience.* (R150)

Prior work has suggested that in designing algorithmic systems, users' uncertainty about the algorithms (while challenging) can lead them to "deeper thinking and even more creative and innovative use of the system" (Eslami et al. 2016). In our study, the users' confusion about their review scores triggered such an innovative use of the system — proding the algorithm to understand its functionality and changing the usual use of a review to reflect it. Reviewers used their review content to raise awareness among other users about a rating bias in the system rather than only sharing their hotel stay experience.

Righting a Wrong

> *This survey calculated its own rating; I would rate it 1.0.* (R40)

Understanding the bias and making other users aware of it was not the only action reviewers took when confronted by their biased review score. Over half of the reviewers (n=95) tried to *correct* the bias. Believing that their review score is "*very exaggerated*" (R60) and "*far too kind*" (R50) for their stay experience, many users disclaimed their score: "*I don't know how it calculated a 2.5 but that's not my rating!*" (R139). They tried to change their review score by announcing their "*true rating*" (R71) in their review text; whether their calculated score was 2.5: "*Definitely not a 2.5 score. 1 would be my true score*" (R119), or higher: "*My rating, as added up by Booking.com of 8.8 is more like a 7 in reality*" (R140). Some asked others to consider their corrected score rather than the one the algorithm calculated: "*For some reason the lowest score I can give them on booking is 2.5. But don't get me wrong, this is not even a 1 out of ten*" (R28).

Users' attempts to correct the rating bias via their review text correspond closely with "improvisation theory," where an unmet system requirement triggers users to improvise, making ad hoc adjustments to their system use to achieve their goal (McGann and Lyytinen 2010). Here, users' improvisation was a "repair" activity. Past work has studied the value of repair in technology reuse and the fixing of broken machines through users' creative and improvisational work (Jackson and Kang 2014; Houston et al. 2016). Our findings point to a broader story: as users try to repair broken devices, they might also try to repair a broken algorithm. And in doing so, they exhibit improvisational and innovative work by changing their review text to reflect their real review score.

Manipulating the Algorithm Users' "repairs" of the biased system did not stop at changing their review texts. A few reviewers aimed to fix the calculated review score itself. To do so, they manipulated the evaluation criteria subscores they had previously assigned to force the algorithm to calculate a final review score closer to their desired score:

After the auto-calculated review score (it runs from 2.5-10, NOT 1-10, so a below average shows up as a 5 rating), I altered my ratings on the staff to show what the overall score should be. (R152).

Although this manipulation might misrepresent a reviewer's opinion about a specific criterion, it did not stop them from changing their subscores. Their desire to match the overall review score with their stay was stronger than their desire to match each subscore. Reviewers lowered the criteria ratings until they were satisfied with the final score:

I had given the hotel the lowest score to customer service and pretty high scores to most other things as the hotel indeed looks very nice and it's clean. But [...] this hotel stay deserved a 3 not an 8 which the algorithm' had automatically calculated on the back of my responses and doesn't allow me to amend. I amended other responses accordingly to reach the 3 the hotel deserved. (R38)

Prior work showed users might manipulate an algorithm's inputs to create workarounds and maintain control over systems such as ride-sharing services (Lee et al. 2015) and social feeds (Eslami et al. 2015; 2016). In our study, however, users do not create workarounds only for self-concerned reasons, but also to correct a bias to help others.

Trust Breakdown

I will never trust your rating again. (R82)

Algorithms are "stabilizers of trust" (Gillespie 2012): they are supposed to be objective and free of intentional bias. But if they are not, how would that affect users' behavior? Confronted by a biased rating, some reviewers (n=26) argued that Booking.com "*is misleading [the] public*" (R9). This belief resulted in a mistrust of the platform. Users stated that giving a hotel a score of 2.5 when they rated every evaluation criteria at its lowest would misguide other readers:

How [are] you calculating the ratings as 2.5 when I rated every option at the lowest. 2.5 is giving them a plug for someone to stay there. STAY AWAY (R73)

Mistrust of the platform led users to call it "*a con*" (R2) and a "*SCAM!!!*" (R6), and warn others that "*the rating by booking.com is rigged*" (R30). They announced their own mistrust, but also suggested others "*NOT TRUST THE CUSTOMER RATINGS ON THIS SITE*" (R152).

Lack of Agency In reviews questioning the algorithm, users rhetorically asked questions like "*Is it possible to create a score lower than 2.5?*" (R59) and "*Don't know what you would have to say to get a 0?*" (R119). Many reviewers tried to achieve lower scores than 2.5, but could not: "*I would give a lower rating if I could, but this review will not allow anything below 2.5*" (R128). Their lack of "*control of the overall review score calculator*" (R96) made them "*VERY UNHAPPY*" (R74). They "*don't like that booking.com [is] taking all [their] lowest score[s] possible and giving an overall rating of 2.5*" (R29). The fact that the algorithm "*forces*" (R141) a score upon them even led one reviewer not to leave any comment about their hotel stay, instead taking their whole review to question their lack of control over their rating:

This review is rigged.[...] I am not commenting as the survey would not allow me to rate the property. The survey already calculated my survey to 7.5 without providing me the ability to input my rating or data. (R153)

Users develop theories about how an algorithm works, but they can only act on those theories where they feel control (Eslami et al. 2016). Once users understand an algorithm, they build workaround strategies to maintain control over it (Lustig et al. 2016; Lee et al. 2015). Having a sense of agency over an algorithmic presentation of their data has led users to have more engaging and trusting interactions with an algorithmic system (Eslami et al. 2015). Losing this sense of agency may result in dissatisfaction or even a breach of trust with the system; particularly if this lack of control has been caused due to a bias, as our results demonstrate.

Questioning the Algorithm, Asking for Change Once their trust was broken, some reviewers challenged Booking.com to explain its algorithm:

Your review ratings make no sense. It seems the lower the client rating the higher you rate yourself. A Complete fraud! You calculate my rating as 7.9 ? My personal opinion is a maximum of 3 out of 10. Please explain. (R83)

These users questioned whether Booking.com's ratings were of any use, if they are biased: "*What is the point of offering customers to leave reviews if you just inflate them yourself and clearly are not interested in the feedback?*" (R39). Some users even requested changes to the algorithm, asking Booking.com to "*gather correct information and post the truth rather than manipulated ratings* (R161). They asked Booking.com to "*stop tricking people into*" (R122) low quality hotels by skewing their ratings:

"Booking.com needs to change the way they do their ratings!" I Want to know how you at booking.com come up with your scores. Do you even read anything people say? Maybe two people said anything a little nice about this place but I think they were being sarcastic....Maybe people should start rating you guys! (R104)

Stating a Departure from the Platform A few reviewers experienced a breach of trust with the whole platform, not only its rating system. Announcing their dissatisfaction with the ratings and asking for explanation and change in the algorithm was not enough. These users stated that they would not use the platform again: *I put the lowest rating possible for this hotel, and booking.com is still giving them a 2.5 rating. I won't use this website again* (R134). Another wrote, "*I don't know if I'd use this service again. For instance, I rated everything super sad face except location. Which I rated sad face and booking.com 'calculated' my rating at 2.9*" (R162). While we do not know if these reviewers stopped using Booking.com, these statements, though extreme, are indicative of the serious effects a biased algorithm can have on the interaction of user and system. When a user finds bias in a part of a platform, she might judge the whole, perhaps even leaving it entirely.

Limitations

While our method of searching for users who mentioned the rating bias in their review was able to detect many, it did not detect them all. However, this was not our goal. We aimed to understand whether there were users who were aware of the bias and if so, how they behaved around the bias. In addition, we only analyzed behavior of users who reflected the bias in their review, and not those who became aware of the bias but did not mention it in their review. As a result, we cannot comment on the number of users who discovered the bias.

We explored users' behavior at a specific time (when users were writing their reviews); we do not know whether and how their behavior might change in the future (e.g., in their next use of the system) as we did not have the means to contact these users. Booking.com user profiles are anonymous. In our study, a fragment of the algorithm's biased outcome was visible to the users via the Web site interface. We investigated users' behavior around this fragment. We look forward to extending this analysis to investigate how users perceive and manage other types of biased algorithms whose bias might not be immediately discernible by users via the interface.

Discussion

From Bias Awareness to Situated Actions

Users usually have plans when they interact with a system; in a hotel rating platform, for instance, a user might intend to compliment excellent hotel staff or to warn others away from a hotel they disliked. Plans change, however, and are difficult to separate from the actions through which they emerge (Suchman 1987): seeing a biased score might lead users to revise their plans. In this study, we built on this theory of situated actions to understand whether and how users orient their actions towards an algorithm that might bias their intended review score. In doing so, users shifted their attention to different aspects of their interaction with the system, depending on what they wished to change.

Focus on Own Experience Some who were unsatisfied with their hotel stay and noticed that their review score did not accurately reflect their experience focused their efforts on ensuring an accurate representation of their hotel stay through the comment option, and not on assessing the rating system itself or informing others about that rating system. To make others aware of how bad their hotel stay was, for example, they noted that their rating was "*way too high*" (R43) and "*this hotel' doesn't deserve that rating or [their] money*" (R124), but did not comment on where the bias might come from, or how such bias might impact other ratings in the system. They mainly focused on correcting how platform presented their own stay experience by informing others that their review score did not reflect their hotel stay correctly.

A Shift in Focus: From Own Experience to the System Others, when encountering a system that did not afford them an opportunity to realize their plans, turned their attention from their hotel stay experience to the system. These users, when confronted by the rating bias, wanted to know how the algorithm worked and wanted to let others know as well. In doing so, some generalized the bias they observed in their own review score to the whole system. They inferred that the algorithm might bias all hotel ratings towards higher scores, suggesting that others ignore the ratings or not trust the entire platform. Users' shift in focus towards algorithms can be used to discover bias or other unexpected outcomes of algorithmic systems, as we discuss below.

Watchdog from within: Bringing Bias to the Surface

In using their reviews to inform others about rating bias on Booking.com, many reviewers employed an audit practice: they detected a bias and tried to publicize it. We foresee many opportunities in designing for this practice as an explicit affordance of platforms wherein bias may unexpectedly emerge, and perhaps even some advantages over other approaches to discovering such biases. Existing algorithm audit techniques have usually been used by third-parties (such as researchers and watchdog organizations) who are neither system designers nor regular users. Audits by third-parties are limited by their lack of direct access to the algorithm at hand, but also by their perspective from outside the actual performance and use of a particular interface in day-to-day experience, or consistent exposure to the algorithm's potential biases. An audit by users, however, looks for bias from the viewpoint of regular use, perhaps increasing the likelihood of detecting bias, and certainly leading to discovery and correction informed by embedded practice. This practice also uses the platform itself to let other users know about the bias, which increases the likelihood that other users will become aware as well.

These benefits call for use of "collective audit" practices in algorithmic systems. Previous work suggested an audit technique in which users come together as volunteers to audit an algorithmic system, in collaboration with researchers (Sandvig et al. 2014). This technique, however, requires more than a user's day-to-day usage to detect bias; it requires users to test a system with different inputs and collaborate with researchers. Similar approaches exist on external websites where users of travel websites come together to understand how an algorithm might work[4]. Yet this practice needs a second platform for users to join together and audit an algorithm. What if a platform could provide users with affordances that they could use to detect and report an algorithmic bias via the platform itself, particularly those biases that the systems designers did not anticipate themselves?

Our study illustrates that users can detect algorithmic bias during their regular usage of a system. It also shows that they do want to inform others about it, if the platform allows. This suggests the potential of "bias-aware design" to aggregate the power of many users and bring algorithm bias to the surface. By bias-aware design, we mean design in which users are able to report algorithmic bias if they notice it. At the simplest level, this can be an interface that allows users to report a bias (e.g., a bias-report button). Achieving a bias-aware design that utilizes the benefits of a collective audit practice efficiently, however, remains for future work.

[4] http://biddingfortravel.yuku.com/

What's Next? (Enough) Transparency and Trust

At first glance, removing an algorithm's bias after detecting it may appear to be the simplest solution. For example, Fandango, a movie rating website, corrected an algorithmic bug that rounded up to a higher rating rather than to the nearest half star (Hickey 2015). Flickr resolved its auto-tagging bias (that labeled Black people with the tag "ape") by manually removing those tags and using those mistakes to improve their algorithm (Hern 2015). These solutions, however, are not always practical. What if an algorithm's bias comes from an inherent bias in input data? What if the bias did not result from a simple software bug but rather from a complex interaction between thousands of parameters in an algorithm?

These challenges call for alternative solutions to confront algorithmic bias. One solution is adding transparency to communicate algorithmic process in the interface. Algorithmic transparency via design can lead users to a more intelligent and adaptive use of a system (Khovanskaya, Bezaitis, and Sengers 2016). We, however, note that making a system completely transparent is usually neither possible nor desired. Algorithms are complex, dynamic, and unpredictable. Even if a designer can gain enough technical literacy to analyze an algorithm, it is often impossible to recreate the complicated and embedded internal processes of an algorithm via design. What we advocate is the study of "actionable transparency" whereby designers with knowledge of their system communicate pivotal algorithmic process cues in the interface — in some cases with features that allow for poking and prodding. Comparison also serves as a powerful means for revealing elements of algorithmic process (Eslami et al. 2015). Such level of transparency can be particularly helpful if an algorithmic system might bias users' experiences. Kulshrestha et al., for example, proposed an alternative design for Twitter search in which users are exposed to Twitter's potential political bias (Kulshrestha et al. 2017).

Such designs might benefit both users and the systems they use. For users, increased transparency in some cases can make them aware of existing biases in an algorithmic system, showing that algorithms are not always free of bias. This awareness can help them to adjust their behavior. Making users aware of bias can benefit a system as well, by building user trust in the system. For instance, analyzing the effects of transparency on an algorithmic peer assessment interface showed that a more transparent grading algorithm increased users' trust in the system when their expectations were violated by receiving a lower than expected grade. To maintain this trust, however, we need balanced transparency; too much may also affect trust (Kizilcec 2016).

On Booking.com, a part of users' distrust comes from the inconsistency between what the interface claims (1) and what the algorithm calculates (2.5) as the lowest review score. Even if Booking.com's business policy prevents modifying its lowest score to 1, making this policy transparent (e.g. by highlighting the 2.5 limit for low scores in the interface) may increase user trust in the system .

Another option for increasing transparency is to provide opportunities for comparison between different algorithms' outputs. For example, consider a design in which a user is able to observe and compare a hotel's rating across different online hotel rating platforms (for example by providing the data we collected in our cross-platform audit technique). This comparison might boost users' understanding of algorithmic systems and their potential biases. Further research, however, is required to explore such methodologies.

Conclusion

We detected and quantified a rating algorithm's bias using a cross-platform audit technique. We found that users oriented towards and adapted their behavior around this bias: they tried to look into the algorithm's black box, correct the bias, and demonstrate a trust breakdown with the platform. These findings bring many opportunities and challenges. Designers of algorithmic systems can use users' collective power in auditing algorithms to shed light on potential existing algorithmic biases. Resolving these biases, however, is not easy. We argue that adding (enough) transparency and "actionable transparency" to algorithmic systems can rebuild users' trust in a system. As more algorithmic processes curate online content, they are more prone to bias user experiences. This work is a first step towards understanding the effects of such biases on user behavior. We hope our findings inspire future designs to better accommodate bias awareness among users.

Acknowledgement

This work was funded by NSF grant CHS-1564041.

References

Anderson, M., and Magruder, J. 2012. Learning from the crowd: Regression discontinuity estimates of the effects of an online review database. *The Economic Journal* 122(563):957–989.

Barocas, S.; Hood, S.; and Ziewitz, M. 2013. Governing algorithms: A provocation piece. *SSRN 2245322*.

Bishop, T. 2015. Amazon changes its key formula for calculating product ratings and displaying reviews. *GeekWire*.

Booking.com. 2015. How is the overall guest review score calculated for my property?

boyd, d. 2014. *It's Complicated: The Social Lives of Networked Teens*. New Haven: Yale University Press.

Chen, L.; Mislove, A.; and Wilson, C. 2015. Peeking beneath the hood of uber. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*.

Datta, A.; Tschantz, M. C.; and Datta, A. 2015. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies* 2015(1):92–112.

Diakopoulos, N. 2014. Algorithmic accountability reporting: On the investigation of black boxes. *Tow Center for Digital Journalism, Columbia University*.

Edelman, B. G.; Luca, M.; and Svirsky, D. 2015. Racial discrimination in the sharing economy: Evidence from a field experiment. *Harvard Business School* (16-069).

Eslami, M.; Rickman, A.; Vaccaro, K.; Aleyasen, A.; Vuong, A.; Karahalios, K.; Hamilton, K.; and Sandvig, C. 2015. "i

always assumed that i wasn't really that close to [her]": Reasoning about invisible algorithms in news feeds. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*.

Eslami, M.; Karahalios, K.; Sandvigt, C.; Vaccaro, K.; Rickman, A.; Hamilton, K.; and Kirlik, A. 2016. First i like it, then i hide it: Folk theories of social feeds. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2371–2382. ACM.

Fowler, G. 2011. Judge dismisses suit against yelp. *The Wall Street Journal*.

Gillespie, T. 2012. The Relevance of Algorithms. *Media Technologies*.

Goodman, B., and Flaxman, S. 2016. Eu regulations on algorithmic decision-making and a right to explanation. In *ICML Workshop on Human Interpretability in Machine Learning*.

Hamilton, K.; Karahalios, K.; Sandvig, C.; and Eslami, M. 2014. A path to understanding the effects of algorithm awareness. In *Proc. CHI EA 2014*, 631–642. ACM Press.

Hannak, A.; Sapiezynski, P.; Molavi Kakhki, A.; Krishnamurthy, B.; Lazer, D.; Mislove, A.; and Wilson, C. 2013. Measuring personalization of web search. In *Proceedings of the 22nd international conference on World Wide Web*.

Hannak, A.; Soeller, G.; Lazer, D.; Mislove, A.; and Wilson, C. 2014. Measuring price discrimination and steering on e-commerce web sites. In *Proceedings of the 2014 conference on internet measurement conference*, 305–318. ACM.

Hern, A. 2015. Flickr faces complaints over 'offensive' autotagging for photos. *The Guardian*.

Hickey, W. 2015. Be suspicious of online movie ratings, especially fandangos. *FiveThrityEight*.

Houston, L.; Jackson, S. J.; Rosner, D. K.; Ahmed, S. I.; Young, M.; and Kang, L. 2016. Values in repair. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 1403–1414. ACM.

Jackson, S. J., and Kang, L. 2014. Breakdown, obsolescence and reuse: Hci and the art of repair. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 449–458. ACM.

Keertipati, S.; Savarimuthu, B. T. R.; and Licorish, S. A. 2016. Approaches for prioritizing feature improvements extracted from app reviews. In *the International Conference on Evaluation and Assessment in Software Engineering*.

Khovanskaya, V.; Bezaitis, M.; and Sengers, P. 2016. The case of the strangerationist: Re-interpreting critical technical practice. In *Proceedings of the ACM Conference on Designing Interactive Systems*, 134–145.

Kizilcec, R. F. 2016. How much information?: Effects of transparency on trust in an algorithmic interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2390–2395. ACM.

Kliman-Silver, C.; Hannak, A.; Lazer, D.; Wilson, C.; and Mislove, A. 2015. Location, location, location: The impact of geolocation on web search personalization. In *the ACM Conference on Internet Measurement Conference*, 121–127.

Kulshrestha, J.; Eslami, M.; Messias, J.; Zafar, M. B.; Ghosh, S.; ; Gummadi, K.; and Karahalios, K. 2017. Quantifying search bias: Investigating sources of bias for political searches in social media. In *the Conference on Computer-Supported Cooperative Work & Social Computing*.

Lécuyer, M.; Ducoffe, G.; Lan, F.; Papancea, A.; Petsios, T.; Spahn, R.; Chaintreau, A.; and Geambasu, R. 2014. Xray: Enhancing the webs transparency with differential correlation. In *23rd USENIX Security Symposium*.

Lee, M. K.; Kusbit, D.; Metsky, E.; and Dabbish, L. 2015. Working with machines: The impact of algorithmic and data-driven management on human workers. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 1603–1612. ACM.

Lustig, C.; Pine, K.; Nardi, B.; Irani, L.; Lee, M. K.; Nafus, D.; and Sandvig, C. 2016. Algorithmic authority: the ethics, politics, and economics of algorithms that interpret, decide, and manage. In *CHI Extended Abstracts*, 1057–1062. ACM.

McGann, S. T., and Lyytinen, K. 2010. Unpacking the dynamics of is user improvisation: A research framework. In *2010 Hawaii International Conference on System Sciences (HICSS)*, 1–10. IEEE.

Muddiman, A. 2013. Searching for the next us president: differences in search engine results for the 2008 us presidential candidates. *Journal of Information Technology & Politics* 10(2):138–157.

O'Neill, M. 2012. Youtube responds to reply girls, changes related & recommended videos algorithm. *Adweek*.

Rader, E., and Gray, R. 2015. Understanding user beliefs about algorithmic curation in the facebook news feed. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, 173–182.

Sandvig, C.; Hamilton, K.; Karahalios, K.; and Langbort, C. 2014. Auditing algorithms: Research methods for detecting discrimination on internet platforms. In *Data and Discrimination*.

Seaver, N. 2013. Knowing algorithms. *Media in Transition* 8:1–12.

Soeller, G.; Karahalios, K.; Sandvig, C.; and Wilson, C. 2016. Mapwatch: Detecting and monitoring international border personalization on online maps. In *Proceedings of the 25th International Conference on World Wide Web*.

Stroud, M. 2014. The minority report: Chicago's new police computer predicts crimes, but is it racist? *The Verge*.

Suchman, L. A. 1987. *Plans and situated actions: The problem of human-machine communication*. Cambridge university press.

Sweeney, L. 2013. Discrimination in online ad delivery. *Queue* 11(3):10.

Tucker, H. 2016. Australian uber drivers say the company is manipulating their ratings to boost its fees. *Businessinsider*.

WhiteHouse. 2016. Big data: A report on algorithmic systems, opportunity, and civil rights. *Washington, DC: Executive Office of the President, White House*.

Yelp. 2010. Yelps review filter explained.