The Foundations of Algorithmic Bias

This morning, millions of people woke up and impulsively checked Facebook. They were greeted immediately by content curated by Facebook's newsfeed algorithms. To some degree, this news might have influenced their perceptions of the day's news, the economy's outlook, and the state of the election. Every year, millions of people apply for jobs. Increasingly, their success might lie in part in the hands of computer programs tasked with matching applications to job openings. And every year, roughly 12 million people are arrested. Throughout the criminal justice system, computer-generated risk-assessments are used to determine which arrestees should be set free. In all these situations, algorithms are tasked with making decisions.

Algorithmic decision-making mediates more and more of our interactions, influencing our social experiences, the news we see, our finances, and our career opportunities. We task computer programs with approving lines of credit, curating news, and filtering job applicants. Courts even deploy computerized algorithms to predict "risk of recidivism", the probability that an individual relapses into criminal behavior. It seems likely that this trend will only accelerate as breakthroughs in artificial intelligence rapidly broaden the capabilities of software.

Turning decision-making over to algorithms naturally raises worries about our ability to assess and enforce the neutrality of these new decision makers. How can we be sure that the algorithmically curated news doesn't have a political party bias or job listings don't reflect a gender or racial bias? What other biases might our automated processes be exhibiting that that we wouldn't even know to look for?

The rise of machine learning complicates these concerns. Traditional software is typically composed from simple, hand-coded logic rules. IF condition X holds THEN perform action Y. But machine learning relies on complex statistical models to discover patterns in large datasets. Take loan approval for instance. Given years of credit history and other side information, a machine learning algorithm might then output a probability that the applicant will default. The logic behind this assessment wouldn't be coded by hand. Instead, the model would extrapolate from the records of thousands or millions of other customers.

On highly specialized problems, and given enough data, machine learning algorithms can often make predictions with near-human or super-human accuracy. But it's often hard to say precisely why a decision was made. So how can we ensure that these decisions don't encode bias? How can we ensure that giving these algorithms decision-making power doesn't amount to a breach of ethics? The potential for prejudice hasn't gone under the radar. In the last year alone, MIT Technology Review [1], the Guardian [2], and the New York

Times [3], all published thought pieces cautioning against algorithmic bias. Some of the best coverage has come from ProPublica, which quantitatively studied racial bias in a widely used criminal risk-assessment score [4].

Each article counters the notion that algorithms are necessarily objective. Technology Review invokes Fred Berenson's assertion that we are susceptible to 'mathwashing'. That is, we tend to (misguidedly) assume that any system built with complex mathematics at its core must somehow be objective, devoid of the biases that plague human decision-making.

Alas, the public discourse rarely throws light on the precise mechanisms by which bias actually enters algorithmic decision-making processes. Tech Review for example, points to the abundance of men working in computer science without explaining how this might alter the behavior of their algorithms. You might think that the bias seeped through via the air filtration system. The Guardian makes a compelling argument that the "recidivism" predictor encodes racial bias, producing evidence to support the claim. But they never discuss how this came to be, describing the algorithms simply as black boxes. Similarly, the New York Times piece calls attention to bias and to the opacity of FaceBook algorithms for new curation, but doesn't elucidate the precise mechanisms by which undesirable outcomes manifest. Admirably, in the ProPublica piece, author Julia Adwin sought the risk-assessment algorithm itself, but software-company Northpointe would not share the precise proprietary formula.

It's encouraging that these pieces have helped to spark a global conversation about the responsibilities of programmatic decision-makers. However, the mystical quality of the discussion threatens to stymie progress. If we don't know how algorithms can become biased, how can we know when to suspect them? Moreover, without this understanding, how can we hope to counteract the bias?

To bring some rigor to the dialogue, let's first run through a crash-course on

what algorithms are, how they make decisions, and where machine learning enters the picture. Armed with this information, we'll then introduce a catalogue of fundamental ways that things can go wrong.

[ALGORITHMS]

To start, let's briefly explain *algorithms*. Algorithms are the instructions that tell your computer precisely how to accomplish some task. Typically, this means how to take some input and producing some output. The software that takes two addresses on a map and returns the shortest route between them is an algorithm. So is the method that doctors use to calculate cardiac risk. This particular algorithm takes the age, blood pressure, smoking status, and a few other inputs, combines them according to a precise formula, and outputs the risk of a cardiovascular event.

Compared to these simple examples, many of the algorithms at the heart of technologies like self-driving cars and recommender systems are considerably more complex, containing many instructions, advanced mathematical operations, and complicated logic. Sometimes, the line between an algorithm and what might better be described as a complex software systems can become blurred.

Consider the algorithms behind Google's search service. From the outside it might appear to be monolithic, but it's actually a complex software system, encompassing multiple sub-algorithms, each of which may be maintained by large teams of engineers and scientists and consisting of millions of lines of code.

There's little that can be said universally about algorithms. Collectively, they're neither racist nor neutral, fast nor slow, sentient nor insensate. If you could simulate your brain with a computer program, perfectly capturing the behavior of each neuron, that program would itself be an algorithm. So, in an important sense, there's nothing fundamentally special about algorithmic decisions. In any

situation in which human decisions might exhibit bias, so might those made by computerized algorithms. One important difference between human and algorithmic bias might be that for humans, we know to suspect bias, and we have some intuition for what sorts of bias to expect.

To dispense with any doubt that an algorithm might encode bias, consider the following rule for extending a line of credit: *If race=white THEN approve loan ELSE deny*. This program, however simple, constitutes an algorithm and yet reflects an obvious bias. Of course, this explicit racism might be easy to detect and straightforward to challenge legally. Deciphering its logic doesn't require formidable expertise.

But today's large-scale software and machine-learning systems can grow opaque. Even the programmer of a system might struggle to say why precisely makes any individual system. For complex algorithms, biases may exist, but detecting the bias, identifying its cause, and correcting may not always be straightforward. Nevertheless, there exist some common patterns for how bias can creep into systems. Understanding these patterns may prove vital to guarding against preventable problems.

[MACHINE LEARNING]

Now let's review the basics of machine learning. Machine learning refers to powerful set of techniques for building algorithms that improve as a function of experience. The field of machine learning addresses a broad class of problems and algorithmic solutions but we're going to focus on supervised learning, the kind directly concerned with pattern recognition and predictive modeling.

Most machine learning in the wild today consists of supervised learning. When Facebook recognizes your face in a photograph, when your mailbox filters spam, and when your bank predicts default risk – these are all examples of supervised machine learning in action.

We use machine learning because sometimes it's impossible to specify a good enough program a priori. Let's say you wanted to build a spam filter. You might be tempted to implement a rule-based system with a blacklist of particularly spammy words. Is any email referring to "Western Union" spam? Perhaps. But even so, that only describes a small percentage of spam. There's still the solicitations from illegal drug companies, pornographic sites, and the legendary Nigerian prince who wants to wire you millions of dollars.

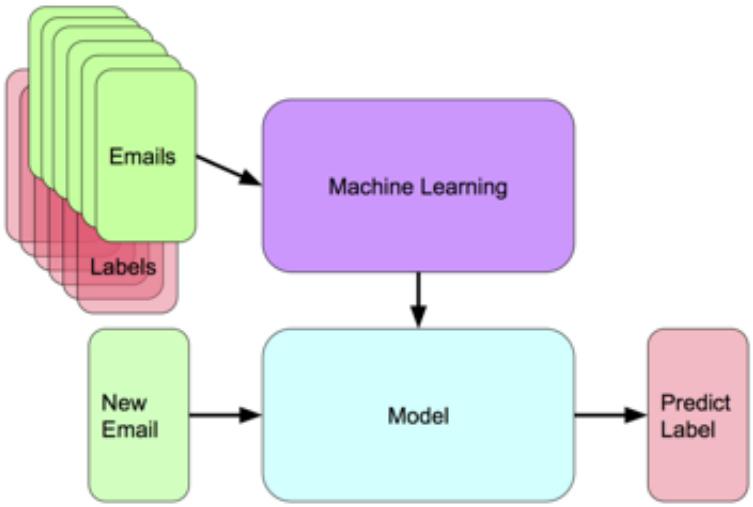
Suppose now that through herculean effort you produced a perfect spam filter, cobbling together 1000s of consistent rules to cover all the known cases of spam while letting all legitimate email pass through. As soon as you'd completed this far-fetched feat and secured some well-deserved sleep, you'd wake up to find that the spam filter no longer worked as well. The spammers would have invented new varieties of spam, invalidating all your hard work.

Machine learning proposes an alternative way to deal with these problems. Even if we can't specify precisely what constitutes spam, we might know it when we see it. Instead of coming up with the exact solution ourselves by enumerating rules, we can compile a large dataset containing emails known either to be spam or to be safe. The dataset might consist of millions of emails, each of which would be characterized by a large number of attributes and annotated according to whether it's believed (by a human) to actually spam or not. Typical attributes might include the words themselves, the time the email was sent, the email address, server, and domain from which it was sent, and statistics about previous correspondence with this address.

Already, you might see a problem. *Who gets to decide which emails are spam and not?* What biases may factor into these decisions? If the labelers think all emails from Nigeria constitute spam, how can we justify using a system that will treat millions of people unfairly?

Once we've got a dataset, we can specify a flexible family of statistical models for

mapping between an email and a probability that it is spam. A simple model might be to assign a score (weight) to every word in the vocabulary. If that weight is positive, then it increases the probability that the email is spam. If negative it decreases the probability.



To calculate the final score, we might sum up the counts of each word, multiplying each by the corresponding weight. This describes the family of linear models, a classical technique from statistics. Machine learning practitioners also have many, more complicated, models at their disposal, including tremendously popular neural networks (a family of techniques now referred to as deep learning). In our present discussion, the exact form of the model doesn't matter much.

When we say the machine learns, we simply mean that as it sees more and more data, it updates its belief, as by tuning the weights, about which model in the family is best. So rather than building a spam filter with a bank of rules such as "IF contains("Western Union") THEN SPAM", we'd curate a large list of thousands or millions of emails, indicating for each whether or not it's spam. These labels are often collected actively, as by crowdsourcing low-wage workers through services like Amazon's mechanical turk. Labels can also be collected passively, as by harvesting information when users explicitly mark

emails as spam or remove emails from their spam boxes to their inboxes.

For all supervised machine learning models, the big picture remains the same. We have a collection of examples of (hopefully representative) data. We also have a collection of corresponding labels collected either actively or passively (typically from human annotators). These reflect an (often subjective) choice over what constitutes the ground truth. Stepping back, we've also made a subjective choice regarding what's worth predicting in the first place. For example, do we ask our annotators to label *spam*, or *offensive content* or *uninteresting content*?

And sometimes, machine learning practitioners formulate problems in such a way that the very notion of ground truth seems questionable. In many applications, researchers classify sentences or documents according to one of several *sentiments*. Other papers break down emotional classification into two dimensions: an *arousal* score and a *valence* score. Whether these simplistic scores can capture anything reasonably related to the ideas indicated by *emotion* or *sentiment* seems debatable.

[BIAS]

We can now begin to demystify the processes by which undesirable biases can infiltrate machine learning models.

[BIASED DATA]

Perhaps the most obvious way that a machine learning algorithm can become compromised is if the underlying data itself reflects biases.

Consider, for example, a model predicting risk of recidivism. The training examples here would consist of past prisoners' records. The corresponding labels would be binary values (1 if they were convicted of another crime, 0 if not). However, these labels themselves can reflect profound biases. For example, an

individual is only convicted of a crime if they are first caught and arrested. But arrest rates reflect well-document racial biases. Thus, black men, in addition to facing a higher probability of incarceration in the first place, could see their misfortune compounded through use of the recidivism predictor.

You might hope that we could get around this problem by withholding sensitive demographic information from the machine learning algorithm. If the model didn't know who was black and who is white, how could it learn to discriminate between the two?

Unfortunately, it's not so simple. Given a rich enough set of features and a rich enough family of models, the machine algorithm deduce race implicitly, and use this information to predict recidivism. For example, zip code, occupation, even the previous crime committed could each leak clues as to the race of the inmate.

Acting upon the biased model's predictions to make parole decisions could in turn perpetuate the cycle of incarceration. The lesson here is that if the purported underlying data is intrinsically biased, we should expect that the machine learning algorithm will produced commensurately biased models.

Another example of machine learning absorbing the biases in training data recently came to attention as researchers at Boston University and Microsoft Research led by Tolga Bolukbasi examined a technique called word embedding [5].

Word embedding is a technique in which each word in a vocabulary is a assigned to a vector. The main idea is that the meaning of each word can be captured by the angle of the vector. These vectors can be used to represent the word when used as input to a machine learning algorithm.

Researchers made waves in 2013 by showing a technique for learning these vectors by choosing the vectors which best predict the neighboring words in a large corpus of data. Serendipitously, the researchers discovered that these

representations admitted some remarkable properties. Among them, the vectors could be used in straight-forward ways to execute analogical reasoning. One now-famous example showed that in this vector space <China> – <Beijing> roughly equals <Russia> – <Moscow>.

Similar examples showed that <king> – <queen> roughly equalled <prince> – <princess>. And some preliminary work showed that these embeddings were sufficiently useful to perform human-level analogical reasoning on standardized tests like the SAT.

In the last three years word embeddings have become a nearly ubiquitous tool at machine learning and natural language processing labs throughout academia and industry. But Tolga Bolukbasi and colleagues showed that in addition to picking up on meaningful semantic relationships, the word embeddings also picked absorbed common biases reflected in the underlying text corpuses.

In one example, they showed that learned embeddings also coded for man - woman ≈ computer programmer – homemaker. Similarly, in the learned embedding space, the occupations closest to "she" were 1. homemaker 2. nurse 3. receptionist 4. librarian 5. socialite 6. hairdresser.

In contrast, the occupations closes to "he" included 1. maestro 2. skipper 3. protege 4. philosopher 5. captain 6. architect.

Bolukbasi and colleagues proposed a method for identifying the subspace of learned embeddings corresponding to gender and correcting for it. However, we should note that this doesn't correct for any of the myriad other potential biases that might lurk within the word embeddings.

The same might be said of humans. We call attention to certain biases, emphasizing them, testing for their existence, and correcting them as best we can. But only by identifying the problem and proposing a test for it can we address

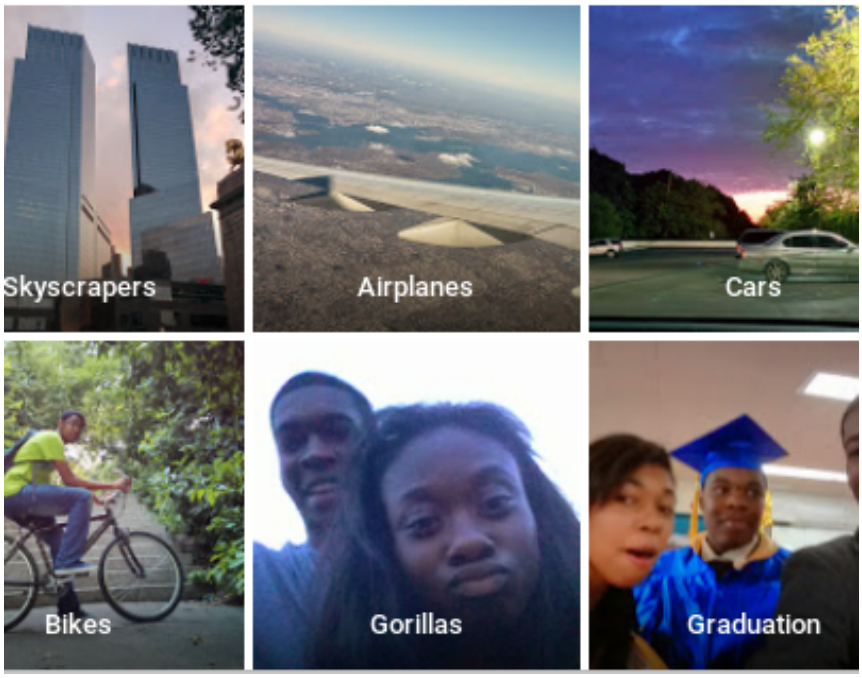
it. It's hard to guess what prejudices might influence human decision-making that we've never thought to examine.

[BIAS by OMISSION]

Even without absorbing explicit biases from datasets, machine learning could produce biased classifications and decisions as a result because the data is implicitly biased by virtue of who is represented and who is omitted.

As glaring example, Google last year added a state-of-the-art objection detection algorithm to its photo app. The algorithm annotated photos with descriptions of the objects they contained such as "skyscrapers", "airplanes", "cars".

However, things went horribly wrong when the app tagged a picture of a black couple as "gorillas".



There a few things to keep in mind here. First, the classifier was likely trained on the academic 1-million image benchmark dataset ImageNet [6], for which the

misclassification rate per 2014 state of the art is 7.4%. That means, for any large population uploading photos, a considerable number will be misclassified.

However, this noted, it's not hard to imagine that being black had something to do with it. To see why, consider the construction of the ImageNet dataset. An academic benchmark, imagenet was built to provide a testbed for advancing computer vision. The dataset contains 1 million images consisting of 1,000 images each from 1,000 object classes. Roughly half of these images depict organisms like humans, birds, and gorillas, while the other half depict artificial objects like airplanes and skyscrapers.

Out of curiosity, I thumbed through the ImageNet explorer, selecting for images of humans and passing over the first 500 by eye. Out of 500 randomly selected images of humans, only 2 depicted black people. These two consisted of one image of Gary Coleman, and another of a black man dressed in drag.

A machine trained on these images might never have seen a typical black man and thus wouldn't know upon seeing one whether to categorize based on color or physiology. Now ImageNet was built by well-meaning academics. It seems exceedingly unlikely that the creators of the dataset intended for models trained on it to misbehave in this fashion.

Released in 2009 by Dr. Fei Fei Li and colleagues, the dataset was inspired that humans see many images per second while forming their ability to recognize objects, and that a computer might need access to a similarly rich dataset.

To my knowledge, the dataset doesn't encode any explicit human bias. There are no images of black men and women mislabeled as gorillas. However, it might alarm us that the absence of blacks from the ImageNet dataset parallels their lack of representation in computer science generally.

While the Google app incident might be isolated and somewhat benign, it's not

hard to imagine how this problem could metastasize. Consider, for example, a security system based on face recognition that only allowed employees to enter a building when it was at least 99% sure of they were correctly ID'd and called security otherwise. If a minority group were missing from training datasets used to train the face recognition algorithm, it might throw alarms disproportionately when these citizens went to work detaining them with greater frequency.

This point is important, even absent racist researchers, corporations or customers, and with algorithms which do not express any intrinsic preferences, absent critical thought we might accidentally birth a system that systematically racially profiles.

On the other hand, we might find this realization empowering because it provides straight-forward prescriptions for how to detect and avoid some kinds of unintentional bias.

Contrary to the Guardian's John Naughton's suggestion that our desire to scrutinize algorithms is stymied by their impenetrable, black-box nature, this particular kind of error can be found simply by examining the training data, a task that surely doesn't require a PhD in machine learning.

[SURROGATE OBJECTIVES]

Thus far we've considered only the ways that bias can infiltrate algorithms via datasets. But this isn't the only way that ethically dubious behavior enters algorithmic decision-making. Another source of trouble can be the choice of objective: What do we choose to predict? And how do we act upon that information?

Consider, for example, the the recommender systems that services like Facebook rely upon to suggest news items from around the world on social media. Abstractly we might state the goal of such a recommender systems is

to surface articles that keep users informed of important events. We might hope that surfaced articles would be truthful. And in an election year, we might hope that different candidates would have equal opportunities to get messages across.

In short, these are the responsibilities we normatively expect humans to take when they make curatorial decisions as with major television stations and newspapers. But this isn't how the algorithms behind real-life recommender systems on the internet work today. Typically, they don't know or care about truth and they don't know about neutrality.

That's not necessarily because internet giants dislike these virtues – it's often simply because it's hard. Where can we find examples of millions of articles scored according to journalistic quality or truth content as assessed by impartial fact-checks? Moreover, ensuring neutrality requires that we not only rank individual articles (and deliver the top ones) but that we rank sets of recommended articles according to their diversity, a considerably harder optimization problem.

Moreover, solving hard problems can be extremely expensive. Google, Amazon, and Facebook have invested billions of dollars in providing machine learning services at scale. And these services typically optimize very simple goals . Solving a yet harder problem with potentially little prospect for additional remuneration cuts against the financial incentives of a large company.

So what do machine learning practitioners typically optimize instead? Clicks. The operating assumption is that people are generally more likely to click on better articles and less likely to click on worse articles. Further, it's easy for sites like Facebook, Google and Amazon to log every link that you click on. This passively collected click data can then be used as supervision to the machine learning algorithms trained to optimize search results. In the end people see more articles that they are likely to click on. The hope would be that this corresponds closely to

what we really care about – that the articles are interesting, or of high quality. But it's not hard to imagine how these goals might diverge. For example, sensational headlines might be more likely to get clicks even if they're less likely to point to true stories.

This is common in machine learning. Sometimes the real problem is difficult to define, or we don't have any solid data. So instead we optimize a surrogate problem, hoping that the solutions are similar enough. And indeed many services, like Google search, despite it's shortcomings, turns up far more relevant results than purely random or chronological selection from the web at large.

But the success of these systems, and our consequent dependence on them, also makes their shortcomings more problematic. After all, no one would be worried about FaceBook's curation of the news if no one received their news from the site.

To see how things could go wrong, we might take a look at the current presidential election. On conventional media like radio and TV, broadcast licensees are required to give equal time to opposing presidential candidates if they request it. That is, even if one candidate might seem more entertaining, or procure higher ratings, we believe that it biases elections for one candidate to receive significantly more coverage than another.

While the adherence of conventional media outlets to this principle might be debatable, it seems clear that denizens of social media were treated to a disproportionate deluge of articles about Donald Trump. While these articles may have truly been more likely to elicit clicks, the overall curated content lacked the diversity we would expect from conventional election coverage.

Of course, FaceBook's news curation is thorny issue. On one hand Facebook has a role in curating the news, even if it doesn't fully embrace its role news organization. On the other hand, Facebook also functions as a public square, a place where people go to speak out loud and be heard. In that context, we

wouldn't expect any enforcement of equal time, nor would we expect all messages to be given equal chance to be heard by all in earshot. But, as we all know, Facebook doesn't simply pass on all information on equally, so it isn't quite a public square either.

It can be hard to anticipate the effects of optimizing these surrogate tasks. Rich Caruana, a researcher at Microsoft Research Redmond presented a compelling case where a predictive machine learning model is trained to predict risk of death in pneumonia patients. The model ended up learning that patients who also had asthma as comorbid condition were given a better probability of survival.

You might wonder why the model reached such a counterintuitive conclusion. The model didn't make an error. Asthma was indeed predictive of survival, this was a true association in the training data.

However, the relationship is not causal. The asthma patients were more likely to survive because they had been treated more aggressively. Thus there's often an obvious mismatch between the problem we want to solve and the one on which we actually train our algorithms.

We train the model to classify asthma risk, assuming nothing changes. But then we operate on the hypothesis that these classifications are causal relationships. Then, when we act based on this hypothesis to intervene in the world, we invalidate the basic assumptions of the predictive model.

As I articulated in a recent paper [7], it's in precisely these situations, where real and optimized objectives disagree, that we suddenly become very interested interpreting models, that is, figuring out how precisely they make decisions. Say, for example, that we want to classify tumors as malignant or benign, and that we have perfectly curated training data. If our algorithm achieves 100% accuracy, then it may not be essential to understand how precisely it makes its decisions. Absent transparency, this algorithm would still save lives. But when our

real-world goals and the optimized machine learning objectives diverge, things change.

Take Facebook's newsfeed as an example. Their real-world goal may be to present a personalized and useful stream of curated content. But likely, the machine learning goal is simply to maximize clicks and/or other superficial measures of engagement. It's not hard to see how these goals might diverge. A story can grab lots of clicks by offering a sensational headline but point to fake news. In that case, the story might be clicky but not useful. Moreover this sort of divergence may be inevitable. There are many situations where for various reasons it might be impossible to optimize real objectives directly. They may be too complex, or there might be no available annotations. In these situations it seems important to have some way of questioning models, either by introspecting them or analyzing their behavior.

In a recent conversation, Rich Caruana suggested a silver lining. These problems may be worse now precisely because machine learning has become so powerful. Take search engines for example. When search engines were predicting total garbage, the salient question wasn't whether we should be following click signal or a more meaningful objective. We simply wondered whether we could make systems that behave comprehensibly at all.

But now that the technology is maturing, the gap between real and surrogate objectives is more pronounced. Consider a spacecraft coming from another galaxy and aiming for earth but pointed (incorrectly) at the Sun. The flaw in its trajectory might only become apparent as the spacecraft entered the solar system. But eventually, as the craft drew closer to the sun, the difference in trajectory would become more pronounced. At some point it might even point in the exact opposite direction.

[DEFINING BIAS]

So far we've punted on a precise definition of bias. We've relied instead on some exemplar cases that seem to fall under a mainstream consensus of egregiously biased behavior. And in some sense, we use machine learning precisely because we want to make individualized decisions. In the case of loan approval, for example, that necessarily means that the algorithm advantages some users and disadvantages others.

So what does it mean for an algorithm to be fair? One sense of fairness might be that the algorithm doesn't take into account certain protected information, such as race or gender. Another sense of fairness might be that the algorithm is similarly accurate for different groups. Another notion of fairness might be that the algorithm is calibrated for all groups. In other words, it doesn't overestimate or underestimate the risk for any group. Interestingly, any approach that hopes to guarantee this property, might have to look at the protected information. So there are clearly some cases in which ensuring one notion of fairness might come at the expense of another.

In a recent paper, Professor Jon Kleinberg gave an impossibility theorem for fairness in determining risk scores. He shows that three intuitive notions of fairness are not reconcilable except in unrealistically constrained cases [8]. So it might not be enough simply to demand that algorithms *be fair.* We may need to think critically about each problem and determine which notion of fairness is most relevant.

[TAKEAWAYS]

Many of the problems with bias in algorithms are similar to problems with bias in humans. Some articles suggest that we can detect our own biases and therefore correct for them, while for machine learning we cannot. But this seems far-fetched. We have little idea how the brain works. And ample studies show that humans are flagrantly biased in college admissions, employment decisions, dating behavior, and more. Moreover, we typically detect biases in human

behavior post-hoc by evaluating human behavior, not through an a priori examination of the processes by which we think.

Perhaps the most salient difference between human and algorithmic bias may be that with human decisions, we expect bias. Take for example, the well-documented racial biases among employers, less likely to call back workers with more more typically black names than those with white names but identical resumes. We detect these biases because we suspect that they exist and have decided that they are undesirable, and therefore vigilantly test for their existence.

As algorithmic decision-making slowly moves from simple rule-based systems towards more complex, human-level decision making, it's only reasonable to expect that these decisions are susceptible to bias. Perhaps, by treating this bias as a property of the decision itself and not focusing overly on the algorithm that made it, we can bring to bear the same tools and institutions that have helped to strengthen ethics and equality in the workplace, college admissions etc. over the past century.

Acknowledgments

Thanks to Tobin Chodos, Dave Schneider, Victoria Krakovna, Chet Lipton, and Zeynep Tufekci for constructive feedback in preparing this draft.

References

1. Byrnes, Nanette, *Why we Should Expect Algorithms to be Biased* 2016 https://www.technologyreview.com/s/601775/why-we-should-expect-algorithms-to-be-biased/
2. Naughton, John *Even Algorithms are Biased Against Black Men* 2016 https://www.theguardian.com/commentisfree/2016/jun/26/algorithms-racial-bias-offenders-florida
3. Tufekci, Zeynep, *The Real Bias Built in at Facebook* New York Times

2016 http://www.nytimes.com/2016/05/19/opinion/the-real-bias-built-in-at-facebook.html
4. Angqin, Julia et al., *Machine Bias* 2016 https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
5. Bolukbasi, Tolga et al. *Quantifying and Reducing Stereotypes in Word Embeddings ICML Workshop on #Data4Good* 2016 https://arxiv.org/abs/1606.06121
6. Deng, Jia, et al. *Imagenet: A large-scale hierarchical image database.* CVPR 2009
7. Lipton, Zachary C., *The Mythos of Model Interpretability.* ICML Workshop on Human Interpretability of Machine Learning 2016) https://arxiv.org/abs/1606.03490.
8. Kleinberg, Jon et al., *Inherent Trade-Offs in the Fair Determination of Risk Scores* https://arxiv.org/pdf/1609.05807v1.pdf

Author: Zachary C. Lipton

Zachary Chase Lipton is an assistant professor at Carnegie Mellon University. He is interested in both core machine learning methodology and applications to healthcare and dialogue systems. He is also a visiting scientist at Amazon AI, and has worked with Amazon Core Machine Learning, Microsoft Research Redmond, & Microsoft Research Bangalore. View all posts by Zachary C. Lipton

Zachary C. Lipton / November 7, 2016 / Machine Learning Ethics / Bias, Machine Learning, Supervised Learning

16 thoughts on “The Foundations of Algorithmic Bias”

Ml student

November 17, 2016 at 5:23 am

Interesting article. One of the examples you're citing (ProPublica) has been refuted by the way. The court software they looked at was NOT biased. The rest of your points still stand of course.

Source: Flores, Bechtel, Lowencamp; Federal Probation Journal, September 2016, "False Positives, False Negatives, and False Analyses: A Rejoinder to "Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And it's Biased Against Blacks."",

URL http://www.uscourts.gov/statistics-reports/publications/federal-probation-journal/federal-probation-journal-september-2016

In fact the ProPublica analysis was so wrong that the authors wrote: "It is noteworthy that the ProPublica code of ethics advises investigative journalists that "when in doubt, ask" numerous times. We feel that Larson et al.'s (2016) omissions and mistakes could have been avoided had they just asked. Perhaps they might have even asked…a criminologist? We certainly respect the mission of ProPublica, which is to "practice and promote investigative journalism in the public interest." However, we also feel that the journalists at ProPublica strayed from their own code of ethics in that they did not present the facts accurately, their presentation of the existing literature was incomplete, and they failed to "ask." While we aren't inferring that they had an agenda in writing their story, we believe that they are better equipped to report the research news, rather than attempt to make the research news." Sounds pretty bad, hu?

Zachary C. Lipton

November 17, 2016 at 3:47 pm

Hi. Thanks for the link. I'll take a closer read and see if the challenge holds water. Regardless of the verdict, I'd caution to be less trusting of anything that just happens to look like a paper. This draft appears in a criminal justice journal,

and is composed by an assistant professor of criminal justice. These are by no means disqualifying, but it's certainly a signal not to take these conclusions on faith.

Regarding the present piece, as you note, the content of the article doesn't hang on the quality of Propublica's piece (it was actually a late addition to the article). For my purposes the very fact that the risk score is used prompts these urgent questions. The matter especially pressing because the score relies on data including years of education and age of first arrest. From multiple angles we ought to be concerned with the ways applying such a model could compromise equal treatment under the law.

static

November 17, 2016 at 1:54 pm

"Unfortunately, it's not so simple. Given a rich enough set of features and a rich enough family of models, the machine algorithm [can] deduce race implicitly, and use this information to predict recidivism. For example, zip code, occupation, even the previous crime committed could each leak clues as to the race of the inmate."

The fact that factors correlated with recidivism may also be correlated with race is irrelevant. What is relevant is whether those factors are correlated with recidivism, and whether they are predictive. E.g., someone unemployed may be more likely to return to a life of petty theft than someone who had particular job skills.

Just because something is correlated with race doesn't mean it is equivalent to using race as a factor in making a decision. It's the exact opposite of that, because people of that race who do not match that correlation do not get treated like those do.

 **Zachary C. Lipton**

November 17, 2016 at 3:31 pm

Hi, I believe you're mistaken on the structure of this argument. Race may indeed be predictive of recidivism. One reason why could be that recidivism is only observed when people are re-arrested. And arrest rates are well-known to be biased higher for some races. In this case, when race is truly predictive (due to the biases of the labels themselves), removing race as a feature doesn't necessarily prevent the model from discriminating.

I hope this allays your confusion. Note, in a kaggle competition, predictive accuracy is enough. But in many situations, there may be models that get high predictive accuracy but should't be used to guide decisions.

 **static**

November 19, 2016 at 5:16 pm

I didn't suggest that "race" would not be correlated with recidivism. As you say, it may be. What I suggested is that removing "race" as an explicit factor DOES make the algorithm no longer "racially stereotyping" or "racially profiling". It is not "deducing race", it is modeling people with features that are not 100% dependent on race. Whether those other factors are 0.2 or 0.8 correlated with whatever racial categorization system one applies to people is irrelevant, as those labeled as being of the race without those features are not affected. The fundamental error of stereotyping, assuming that ALL of class A have feature B because SOME of class A have feature B is the one you are making in your recommended approaches for assessing bias, not the one the algorithm is making in ignoring race.

Zachary C. Lipton

November 19, 2016 at 7:33 pm

I'm glad we're having this discussion. But I have to point out that your analysis is incorrect. Easy example: Imagine that the real police have a strong racial bias and arrest more black people. Thus the rate of recidivism will be higher among black citizens (because recidivism is only measured if you are re-arrested). In this case there would be causal connection between being black and getting arrested. The mechanism for this connection is racism.

Now you could remove the race feature from your data. But say your data contained features on zip-code, job title and income. These features may have nothing to do with crime intrinsically, but they could effectively recover the signal from the missing race feature. So if the labels are truly racist, we should expect the trained model to be racist (even if we eliminate the race feature).

bryan

June 11, 2017 at 2:56 am

Hi Zachary. I'm late to this debate, having only come across this article recently for a class I'm now taking. I enjoyed reading it, and the portions on omission errors was particularly interesting.

I'm also glad there was an actual reasoned, non-personal discussion on the bias issue. This comment is meant in that vein and tone.

I'm having a hard time following your argument. Your beef seems to be more whether or not we should try to predict recidivism based on past history. You believe there is racism at play (a moral issue) in who gets

arrested.

Should such a model be forbidden from taking into account gender(men tend to get rearrested at a much higher rate)? Or age (since younger people tend to commit more violent offenses at all levels)? Are models that do sexist or ageist? Or do those factors reflect reality? Or both?

I don't like the recidivism reality anymore than you – I hope no one does. And I think most thinking people agree there are a wide range of factors at play to explain that reality – racism one among many. But trying to massage the data to get a less robust answer (which incidentally has moral value, is to try to protect the average citizen of any race from becoming a victim of a criminal) because reality makes us queasy isn't science. Not real science, anyway.

Very thought provoking article though.

Zachary C. Lipton

June 11, 2017 at 6:44 am

Thanks for you thoughts. I think what you may be missing is that while inferring the conditional probability P(Y|X) might be a "science", deciding what to model (What is Y?) and how to take actions (as by making sentencing decision) based on this information are not sciences, they are value judgments.

On the specific case of recidivism prediction I think it's an inherently flaws way to be making sentencing decisions. Punishments should be for crimes committed, not future crimes forecasted by flawed models.

Regarding the issue of arrests: yes, there is racism in who gets arrested. So

we don't actually observe an unbiased sample of who commits a crime, we only see who both commits a crime AND gets caught. Because who gets arrested reflects race-based policing, we wind up with a biased sample. The big issue here (ignoring for the moment the fundamental problems with risk-based sentencing in the first place) is that the model is misrepresented as telling us who's likely to reoffend, but actually reflects a different quantity that happens to be biased (in both the statistical and common use sense) against some minorities.

Kim Woods

May 6, 2018 at 12:21 am

It's a shame people seem to be misunderstanding your point here. It's a very important point.

Here is another way of looking at this issue (and yes it's a data labelling issue, so let's take race out of it for now).

Person 1 grew up in a poor family in a deprived area of town. Their first offence was at 15 with some associates stealing a car. Due to age, they were not given a custodial sentence. Their second offence was breaking and entering at age 16, for which they were also caught and given a short custodial sentence in a juvenile facility.

Person 2 is from a fairly wealthy family, and at 28 lives alone, no partner no children. They run a real estate company. For the past three years they were engaged in white collar crime, until this was discovered and, as a result of a good lawyer, paid a fine. Since this they started a new company and have continued with similar white collar crime the past two years as well as occasionally selling Class A drugs on to friends in their network.

> In our data, Person 1 is labelled as having higher recidivism. Some people may have a lower chance of getting caught, but this does not mean they didn't commit crime. Ideally we want future algorithms to detect the crime that currently flies under the radar, not enhance it's ability to stay buried.

Pingback: [Machine Learning Meets Policy: Reflections on HUML 2016 – Approximately Correct]

nicolas

July 3, 2017 at 8:25 am

You shouldn't kept the gorilla example unless your a racist.

Zachary C. Lipton

July 20, 2017 at 9:37 am

> With respect, I don't think you've thought this comment through. The gorilla example presents clear evidence of the the *real* problem that algorithms can inadvertently exhibit racist behavior. How can the community correct the problem without discussing it openly and presenting evidence of its existence?

Vladimir

September 3, 2017 at 2:41 pm

Hello Zachary!
Thank you for a very interesting and insightful article!

As a takeaway I was thinking: "Hey! weren't machines supposed to diminish all

those tools and institutions in the first place, greatly improve time for decisions, and a-priori considered as most impartial?" While the article suggests to treat bias as a property of decision itself. Further implies to treat machine decisions as any other "blackbox" decisions and therefore apply institutions and tools same as for humans!
I believe in that case the institutions are a subject to great transformations.

Ven Kumar

February 5, 2018 at 11:13 pm

Very interesting discussions on bias. is there a 'Fairness' test similar to a "Turing" test that is available or the academia is working on to certify algorithms for their fairness. Is it almost like a Fairness score of of 1 to 10 that can be created .

andrew lubsey

June 11, 2018 at 5:59 am

Zachary . thanks for bringing attention to such an important issue.

Stella

September 11, 2018 at 4:20 am

Based on the previous comments, I'm guessing some people will still not believe, or be able to parse, how a recidivism ML algorithm becomes racist without labels of skin color, and how that might play out in the real world.

Let's take one young black man as an example, who got caught stealing candy once in the city and was turned into the police (instead of a stiff warning from the shopkeep like a white kid might get – one way racism enters the algorithm). In

the local system, black men from the poor part of town are more likely to get picked up, arrested or accused, because racism. It is also because desperateness and glamorization of crime affects poor communities of color, which can lead to more actual crimes being committed by people there. So, the numbers in the ML algorithm are not in the post code's favor. As explained in above article, the algorithm will take in data like location of crime and kid's home (this, as well as being booked for something rather petty, feeds the algorithm racism), crime type, age, level of education, maybe medical history?, previous record etc, but (hopefully) not the way they look. The candy-stealing kid, who is the nicest boy you'd ever meet, maybe just had an urge to test a system (like all teens do). He gets marked as someone who will commit a crime again because his data points are closer to other recidivists (ie black/from a certain area/education level) than perhaps white folk in another part of town.

tl;dr, a well-meaning kid is marked as a criminal for life just because a shop keeper feared him and he lived in a rougher black neighborhood.