Algorithms are everywhere, but what will it take for us to trust them?

July 31 2019, by Kate Smith-Miles

Credit: AI-generated image (disclaimer)

The role of algorithms in our lives is growing rapidly, from simply suggesting online search results or content in our social media feed, to more critical matters like helping doctors determine our cancer risk.

But how do we know we can trust an algorithm's decision? In June, nearly 100 drivers in the United States learned the hard way that sometimes algorithms can get it very wrong.

Google Maps got them all stuck on a muddy private road in a failed detour to escape a traffic jam heading to Denver International Airport, in Colorado.

As our society becomes increasingly dependent on algorithms for advice and decision-making, it's becoming urgent to tackle the thorny issue of how we can trust them.

Algorithms are regularly accused of bias and discrimination. They have attracted concern from US politicians, amid claims we have white men developing facial recognition algorithms trained to work well only for white men.

But algorithms are nothing more than computer programs making decisions based on rules: either rules that we gave them, or rules they figured out themselves based on examples we gave them.

In both cases, humans are in control of these algorithms and how they behave. If an algorithm is flawed, it's our doing.

So before we all end up in a metaphorical (or literal!) muddy traffic jam, there is an urgent need to revisit how we humans choose to stress-test those rules and gain trust in algorithms.

Algorithms put to the test, kind of

Humans are naturally suspicious creatures, but most of us can be convinced by evidence.

Given enough test examples—with known correct answers—we develop trust if an algorithm consistently gives the correct answer, and not just for easy obvious examples but for the challenging, realistic and diverse examples. Then we can be convinced the algorithm is unbiased and reliable.

Sounds easy enough, right? But is this how algorithms are usually tested? It's harder than it sounds to make sure that test examples are unbiased and representative of all possible scenarios that could be encountered.

More commonly, well studied benchmark examples are used because they are easily available from websites. (Microsoft had a database of celebrity faces for testing facial recognition algorithms but it was recently deleted due to privacy concerns.)

Comparison of algorithms is also easier when tested on shared benchmarks, but these test examples are rarely scrutinized for their biases. Even worse, the performance of algorithms is typically reported on average across the test examples.

Unfortunately, knowing an algorithm performs well on average doesn't tell us anything about whether we can trust it in specific cases.

It's not surprising to read that doctors are skeptical of Google's algorithm for cancer diagnosis, which offers 89% accuracy on average. How does a doctor know if their patient is one of the unlucky 11% with an incorrect diagnosis?

With increasing demand for personalized medicine tailored to the individual (not just Mr/Ms Average), and with averages known to hide all sorts of sins, the average results won't win human trust.

The need for new testing protocols

It's clearly not rigorous enough to test a bunch of examples—well-studied benchmarks or not—without proving they are unbiased, and then draw conclusions about reliability of an algorithm on average.

And yet paradoxically this is the approach on which research labs around the world depend to flex their algorithmic muscles. The academic peer-review process reinforces these inherited and rarely questioned testing procedures.

A new algorithm is publishable if it's better on average than existing algorithms on well-studied benchmark examples. If it's not competitive in this way, it's either hidden away from further peer-review scrutiny, or new examples are presented for which the algorithm looks useful.

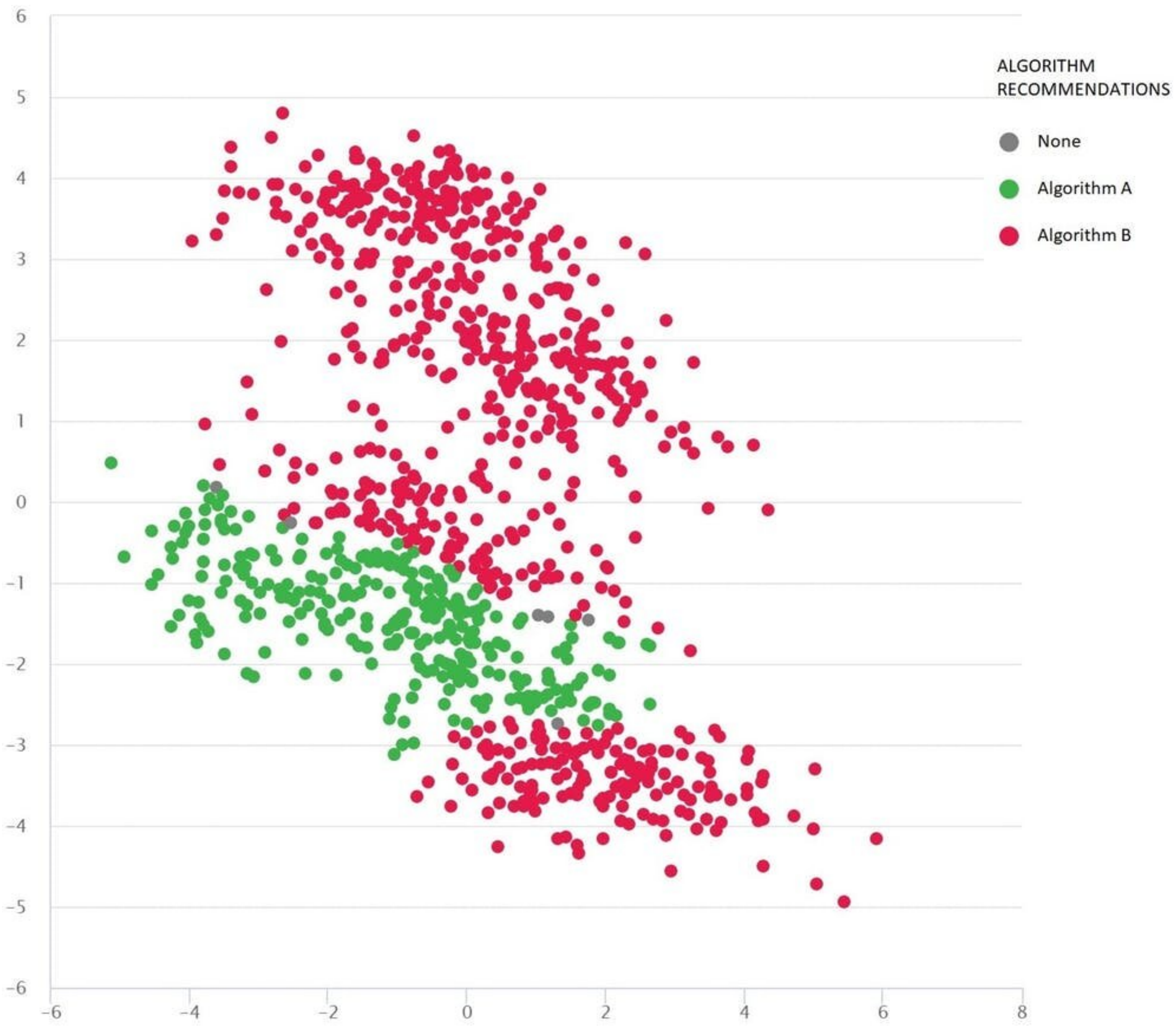
In this way, a warm, flattering light is shone on each newly published algorithm, with little attempt to stress-test its strengths and weaknesses, and present it warts and all. It's the computer science version of medical researchers failing to publish the full results of clinical trials.

As algorithmic trust becomes more crucial, we urgently need to update this methodology to scrutinize whether the chosen test examples are fit for purpose. So far, researchers have been held back from more rigorous analysis by the lack of suitable tools.

We've built a better stress-test

After more than a decade of research, my team has launched a new online algorithm analysis tool called MATILDA: Melbourne Algorithm Test Instance Library with Data Analytics.

It helps stress-test algorithms more rigorously by creating powerful visualizations of a problem, showing all scenarios or examples an algorithm should consider for comprehensive testing.



A Google-maps-type problem with diverse test scenarios as dots: Algorithm B (red) is best on average, but Algorithm A (green) is better in many cases. Credit: MATILDA, Author provided

MATILDA identifies each algorithm's unique strengths and weaknesses, recommending which of the available algorithms to use under different scenarios and why.

For example, if recent rain has turned unsealed roads into mud, some "shortest-path" algorithms may be unreliable unless they can anticipate the likely impact of weather on travel times when advising the quickest route. Unless developers test such scenarios they'll never know about such weaknesses until it is too late and we are stuck in the mud.

MATILDA helps us see the diversity and comprehensiveness of benchmarks, and where new test examples should be designed to fill every nook and cranny of the possible space in which the algorithm could be asked to operate.

The image below shows a diverse set of scenarios (dots) for a Google Maps type of problem. Each scenario varies conditions—like the origin and destination locations, the available road network, weather conditions, travel times on various roads—and all this information is mathematically captured and summarized by each scenario's two-dimensional coordinates in the space.

Two algorithms are compared (red and green) to see which can find the shortest route. Each algorithm is proven to be best (or shown to be unreliable) in different regions depending on how it performs on these tested scenarios.

We can also take a good guess at which algorithm is likely to be best for the missing scenarios (gaps) we haven't yet tested.

The mathematics behind MATILDA helps to create this visualization, by analyzing algorithm reliability data from test scenarios, and finding a way to see the patterns easily.

The insights and explanations mean we can choose the best algorithm for the problem at hand, rather than crossing our fingers and hoping we can trust the algorithm that performs best on average.

By rigorously stress-testing algorithms in this way—warts and all—we should reduce the risk of rogue algorithm decisions, securing the trust of Mr/Ms Average, and perhaps even the most sceptical humans.

This article is republished from The Conversation under a Creative Commons license. Read the original article.

Provided by The Conversation

Citation: Algorithms are everywhere, but what will it take for us to trust them? (2019, July 31) retrieved 10 April 2024 from https://techxplore.com/news/2019-07-algorithms.html