Algorithms and The Law

Deven R. Desai

Georgia Tech Scheller College of Business

FEAR and OPACITY

Critics' Premise

Algorithms combined with data analytics have taken center stage and now "are used to make decisions for us, about us, or with us," in sensitive and subjective areas

- Health-care,
- Employment,
- Credit,
- National Security,
- Networked Devices,
- News, and
- More

Concerns

Power

Structure of Society

Fairness

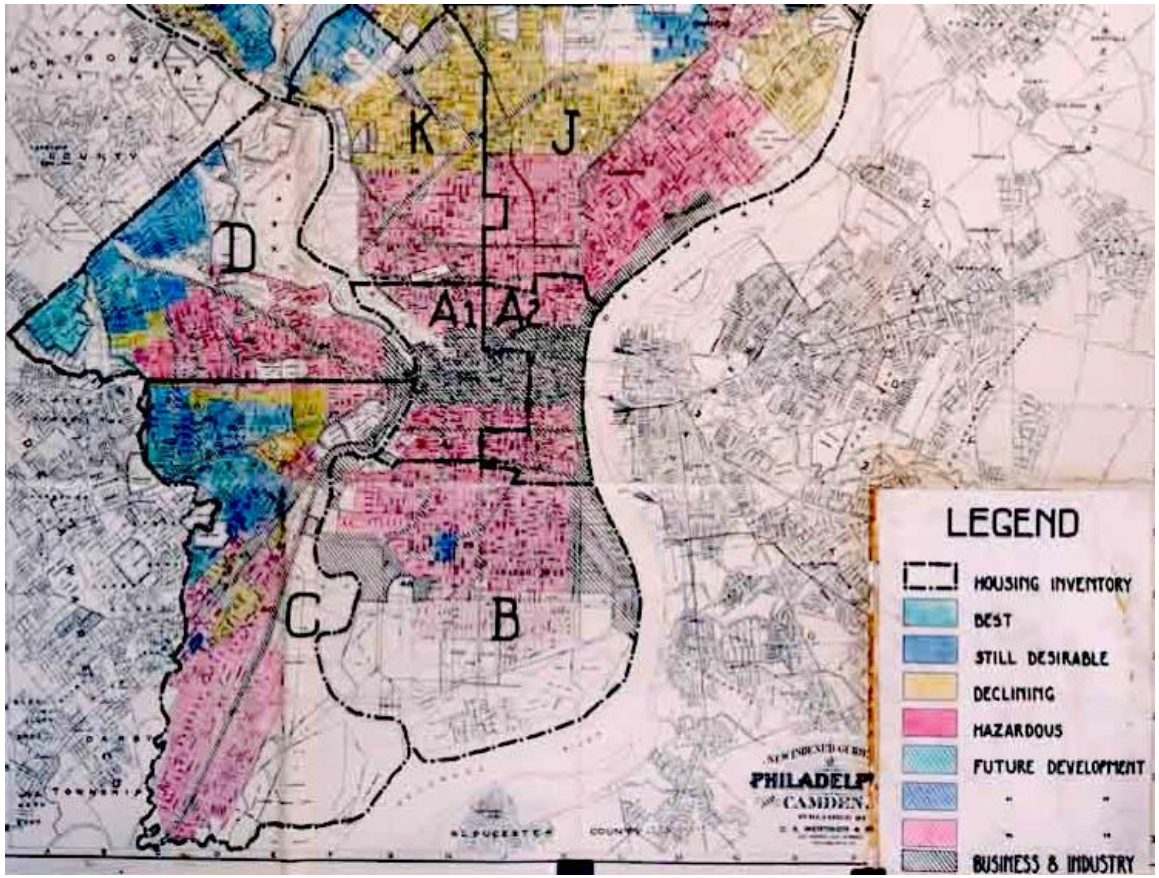
Welfare

Are the Concerns New?

Papirius Praetextatus Entreated by his Mother to Disclose the Secrets of the Deliberations of the Roman Senate

Critics' Premise – *Maybe* A New Problem

Exemplary Issues
Use Code to Discriminate, Suppress Speech, Engage in other Prohibited Acts



A HOLC 1936 security map of Philadelphia showing redlining of lower income neighborhoods. Households and businesses in the red zones could not get mortgages or business loans.

Solution: Transparency!

Solution: Transparency?

The general idea that algorithmic systems are powerful and opaque has led to claims “that virtually any algorithm may deserve scrutiny,” **but consensus on**

- what sort of scrutiny is needed,
- whether different areas affected by algorithms require different solutions, and
- whether algorithms, other factors, or both are the cause of the claimed problems,

is lacking.

Against Transparency

From a technical perspective exposing algorithms to the sun will not only fail to deliver critics' desired results but also may create the illusion of clarity in cases where clarity is not possible.

Roadmap

- Algorithms
- Solutions
- Challenges

Algorithms do not = Magic

Title: The Adoration of the Golden Calf
License: Public Domain
Source: WikiMedia
Author: Nicolas Poussin, (1633)

"The next time you hear someone talking about algorithms, replace the term with "God" and ask yourself if the meaning changes. Our supposedly algorithmic culture is not a material phenomenon so much as a devotional one."

– Ian Bogost, *The Atlantic* (January 2015)

Algorithms Basic Points

- An algorithm is a step-by-step process and “each of the steps must be precise, requiring no human intuition or guesswork.”

Algorithms Not All the Same

- Steps of brushing teeth qualify yet
- Humans “might be able to tolerate it when an algorithm is imprecisely described, but a computer cannot.”
- As Thomas Cormen puts it, “We want two things from a computer algorithm: given an input to a problem, it should always produce a **correct solution** to the problem, and it should **use computational resources efficiently** while doing so.”

Correctness

- Correctness does not work the way policy critics would like.
- An algorithm is correct depending on a specification.

Example

- Consider an algorithm to sort a set of numbers in ascending order.
- An algorithm that shuffled the numbers 5, 3, 4, 2, 1 would sometimes deliver the correct outcome, but it would not be a correct sorting algorithm.
- Even a broken clock is correct twice a day.

Example

- In another case, the algorithm may be able to sort a set of numbers but crash when a duplicate number is entered.
- Thus the algorithm may handle 5, 3, 4, 2, 1 and sort it to return 1, 2, 3, 4, 5. But on the input 5, 3, 4, 1, 1, the algorithm would crash and provide no output.
- This program is not a correct sorting function, and yet it also never returns an incorrect output.

Practical Wall
What is Correct for Policy?

To ask that an algorithm should not “discriminate” or yield some other result prohibited by legal rules, requires that a precise statement, or specification, be provided so that the request is workable for computer scientists.

Surprise to Some

We can posit that both a precise specification and a complete system are available and find that nonetheless it is impossible to test whether the system will yield a certain outcome or even did yield certain outcomes that have already happened, a basic function of all auditing tools.

Surprise to Some

Thus we can see that the idea of testing an algorithm to show that it does not “discriminate” or yield some other result prohibited by legal rules is unworkable.

Specific Limits

- Treat unlawful discrimination as crashes—problems any programmer wishes to avoid
- The cause behind the crash, the bug, can be controlled for, but it is still impossible to catch all bugs.

Irony

- When asked to catch things which might lead to a crash, we reach the sort of precision that the physicist, mathematician, or logician seeks, but *the outcome is that we can show that we cannot show certain things*.
- In the specific case of software, because detecting a potential crash is an undecidable problem, “it is provably impossible for any software-checking tool to detect all possible crashes in all programs.”

Not a Free Pass

- An external, general way (a “transparency tool”) to test may not be available but
- Computer science has rich tools to mitigate the problem (Kroll et. al 2016)

Ways to Mitigate

- We can ask for a commitment or guarantee that certain software was built a specific way and wish to verify that promise. →
- Requires that one starts with or builds programs that are analyzable.
- Then computer science can offer ways to give a 100% guarantee that something is true about a piece of software under certain circumstances.

Ways to Mitigate: Example

- To review an action after-the-fact, you need an audit log.
- We can use cryptographic commitment and zero-knowledge proofs to know that the audit log corresponds to what actually happened

Ways to Mitigate: Example

A passive, outside observer

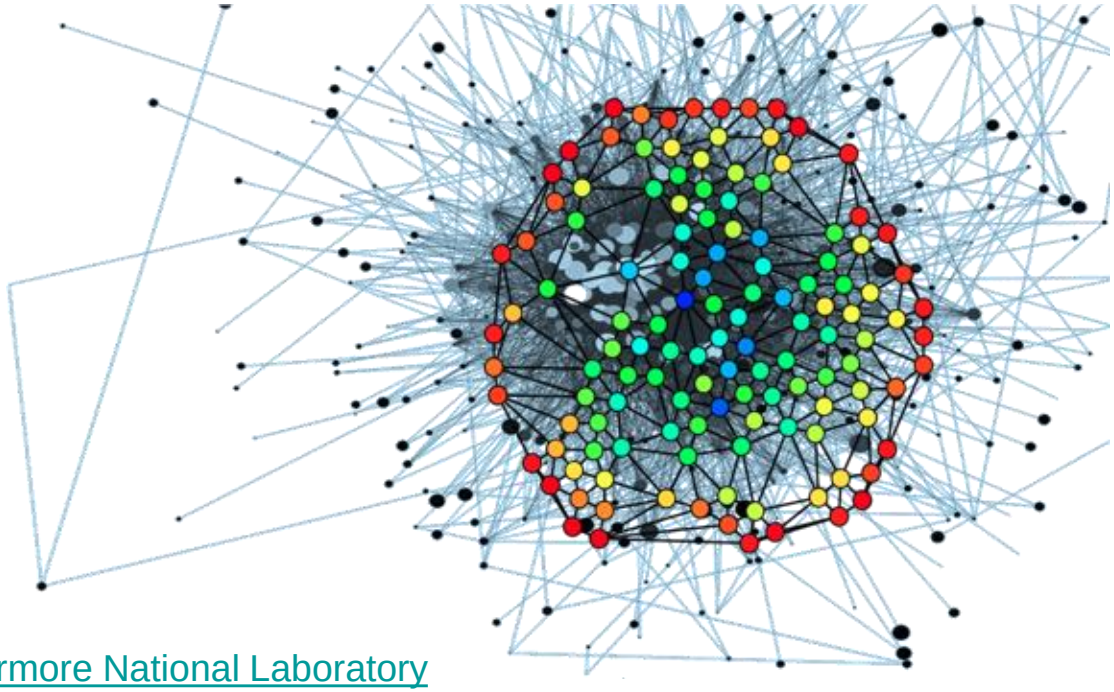
- Can see that the audit log is correct at the time it is created (i.e., at the time the decision is made), and
- Can see that the audit log corresponds to a process with certain desirable properties, such as the same decision policy was applied in all cases.

Limits

- What to do about systems already in place but not designed for evaluation?
  - Already built system (Search, spam filters, ads delivery, social media feeds)
  - Unknown source (malware)

Challenges – Dynamic Systems

- What to do about systems that are dynamic and change over time?



Source: Lawrence Livermore National Laboratory
Title: Image for Technical Focus Area: Machine Learning and Pattern Analysis
Author: U.S. government

Questions

- Does the nature of learning systems mean we will only allow them to be used for non-sensitive areas?
- Are the CS techniques for fairness (e.g. randomization) inherently in tension with certainty or pre-commitment that may be needed by law and policy for auditing?

Specific Questions

- Does it matter whether a public or private entity is using an algorithmic process of concern?
  - Government Visa Lottery
  - Online Ads
  - Engine performance (e.g., VW)
  - Cell phone battery and signal strength (e.g. Apple)
  - Edge Case: Credit system

Conclusion

Together CS and law and policy need to adapt governance to accommodate dynamic systems while still mitigating if not preventing undesired outcomes.

Conclusion

- Must understand that algorithms and data science are not scary, black magic but need to be better explained by the CS community
- Where possible, law and policy needs to offer concrete explanations and stipulations of when regulation will apply and what are prohibited outcomes
- By better working within the technology to be regulated, we will pose problems to solve rather than solutions that will be rejected, or worse provide false comfort while missing the practice at issue