INTERNATIONAL COMPUTATIONAL LAW JOURNAL

CODEX SOURCE ARCHIVE

A Perspective on Legal Algorithms

Alex Pentland

Computational Law Report · Release 1.0 · 2019-12-06

Reader edition prepared by The Research University under CC BY 4.0. This cover was added; the original PDF pages follow unchanged. Original authorship, publication identity and source links are preserved. This independent republication is not sponsored or endorsed by Stanford University, Stanford Law School, CodeX or MIT.

Volume 1 · Issue 1 · Foundations: Law Becomes Computable

Original source and license evidence are recorded on the article landing page.

MIT Computational Law Report

A Perspective on Legal Algorithms

Alex "Sandy" Pentland[1]

[1]MIT

Published on: May 25, 2020

License: Creative Commons Attribution 4.0 International License (CC-BY 4.0)

Legal Algorithms

Code is law, and law is increasingly becoming code. This change is being driven by the growing need for access to justice and the ambition for greater efficiency and predictability in modern business. Most laws and regulations are just algorithms that human organizations execute, but now legal algorithms are beginning to be executed by computers as an extension of human bureaucracies. Already, computer tools are commonly used to help humans make legal determinations in areas such as finance, aviation and the energy sector, most of the logic is computerized and subject only later to human oversight. Even court proceedings are becoming increasingly reliant on computerized fact discovery and precedent, which will likely lead to more and more cases being settled out of court. Moreover, the execution of legal algorithms by computers is likely to dramatically expand as digital systems become more ubiquitous.

As evidenced by the interest and engagement that young lawyers and visionary legal scholars have shown, the legal profession is quietly seizing upon the opportunities provided by the transition to computer-aided human legal practice. It may surprise readers to learn that several law schools have established entrepreneurship programs and incubators focused on legal technology, including Suffolk University Law School and Brooklyn Law School. Faculty of both law schools are among the founders of this Computational Law Report. Young lawyers in training are similarly engaged. I was pleasantly surprised to see that the recent " Open Media Legal Hackathon," organized by the founders of this Report, was hosted by legal academics, technologists, entrepreneurs, community organizers, and others across multiple continents. The legal profession is beginning to go fully digital!

Nevertheless, as legal algorithms transition to being executed by computers, we must be careful not to lose the guardrails of human judgment and interpretation to ensure that the legal algorithms improve justice in our society. We must continue to safeguard, and even substantially increase, human oversight of our legal algorithms.

We must also recognize that current legal and regulatory systems are often poorly designed or out-of-date. As we transition to computer execution of legal algorithms, we have a unique opportunity to make laws more responsive and precise. Relatedly, we should recognize that many legal algorithms fail to achieve their intended aims, or have unintended consequences, and we must ask if there is a better method of ensuring the performance and accountability of each legal algorithm.

Computational Law

How can we achieve greater oversight and accountability of legal algorithms while harvesting their potential to provide greater efficiency, ease of access, and fairness? The obvious answer is to learn from the human-machine systems framework which has evolved over the last century to become the

standard practice in designing and fielding of human-machine systems across the world. Leading examples of this framework include Amazon's fulfillment and delivery systems and internet connectivity systems.

The stunning efficiency and reach of these systems comes, perhaps surprisingly, from modesty: the idea that you can't ever build human-machine systems that "just work." Instead, you will have to continually tweak, reiterate, and redesign them. Once you accept the limitations of the human intellect, you realize that the system must be modular, so you can revise the algorithms easily; the system must be densely instrumented, so you can tell how well each algorithm is working; and, less obviously, the design of the system and each of its modules has to be clearly and directly connected to the goals of the system so that you know what modules to redesign when things go wrong and how to redesign them.

To be clear: some "modules" are software, but others are people or groups of people, all working to execute the algorithms that make up the human-machine system. "Redesigning" human "modules" means reorganizing and perhaps retraining the people, a process familiar as "Kaizen" in manufacturing and as "Quality Circles" in business generally. Note that for the quality circle process to work, the people in the system must clearly understand their connection to the overall goals of the system.

A key element of this design paradigm is testing. We simply cannot design a complex human-machine system that works without extensive testing, field piloting, and evaluation. Testing always begins with a simulation of key components, then the entire system, and concludes with pilot deployments with representative communities as an experiment in which participants give informed consent. Moreover, this testing and evaluation is not just as part of creating the system, it must also happen continuously after large-scale deployment of the system. Things change, and in order to adapt, we must continue to tweak and reengineer the system.

The ability for workers (or regulatory staff members) to critique and revise their jobs (e.g., the Quality Circle process) is key to the success of the overall system. In traditional legal systems, the task of auditing and revising modules based on performance feedback is the role of senior regulators and the courts. The task of auditing and revising the overall system architecture is traditionally the role of legislators.

When the legal system process is compared to more successful human-machine systems it becomes clear that our current legal processes give insufficient thought to instrumenting modules (e.g., why did it take a decade to evaluate broken windows policing?), and insufficient thought to designing systems that are modular and easy to update (e.g., the health care system or tax code). A subtler problem is

that the current legal algorithms are insufficiently clear about the goals they are intended to achieve, and about what evidence can be used to evaluate their performance.

Simple Examples of Computational Law Systems

Some simple examples of using this design framework to build successful legal algorithms may help illustrate these ideas. The first example is a government setting up an automatic, algorithmic legal system -- specifically a traffic congestion taxation system. This system, implemented in Sweden, reads car license plates and charges drivers for use of roads within Stockholm. We can see each of the components of proper legal algorithm design in the Wikipedia description of the system.

- The motivation of the congestion tax was stated as the reduction of traffic congestion and the improvement of certain air quality metrics in central Stockholm. Consequently, the ***goals*** of the system were clear, and the ***measurement criteria*** for system performance were well understood.
- Following seven-months of ***testing*** during a trial period, the tax was implemented permanently.
- After initial deployment, the system design was ***adapted and revised*** to obtain better performance by charging higher prices for the most central part of Stockholm.
- The system was ***audited*** for the first 5 years of operation and demonstrated a decrease in congestion, with some motorists turning to public transport.

While the elements of algorithmic design may seem quite obvious in this example, such considerations are often not present in the creation and operation of algorithmic legal systems. Sweden's congestion tax system has since been used as a model by city governments and urban planners around the world.

The second example is commercial and drawn from my personal experience helping guide Nissan to create an autonomous driving system for their cars. This system design is now the largest deployed autonomous driving system in the world (at level 2). The development of the system began with specifying the design objective:

- The ***goal*** of the car navigation system should be to achieve safer driving without distracting the driver. It should feel like you are just driving the car as usual, but the car just naturally does "the right thing." The human is always fully engaged and in charge.
- Laboratory ***testing*** of the system revealed that the car's idea of "what to do" must match the judgment of human drivers, so that the car never does anything the driver does not expect or understand.
- The system was ***adapted and revised*** through pilot deployments that determined when the car could usefully help the driver, and when it should not try to help. The system was also improved iteratively as new sensing technologies became available.
- Following commercial deployment, the system has been continuously ***audited*** for safety and customer satisfaction, and is continuously updated.

The consequence is that driving has become much safer, and people love the system... although sometimes they fail yo appreciate just how much the system is doing. For instance, drivers often fail to appreciate how the system subtly teaches them to be better drivers. Instead of functioning merely as a tool that replaces humans or human reasoning, these types of systems are more akin to training wheels or guide rails. In fact, the original name for the system was "magic bumper."

Missing Components of Successful Computational Law

Unfortunately, several of the elements highlighted above are underdeveloped or even missing from current legal and regulatory system processes. These include: specification of system performance goals, measurement and evaluation criteria, testing, robust and adaptive system design, and continuous auditing.

Specification of system performance goals

The creation of a new system of legal algorithms (e.g., a law and associated regulation) requires a debate among citizens and legislators concerning objectives and values which results in a clear specification of the overarching goals of the systems' objectives. The failure to specify objectives increases the likelihood that the resulting legal systems will fail to provide good governance and may produce negative unintended consequences.

Measurement and evaluation criteria

To have any chance of determining whether or not something is a success, we need to have an appropriate point of comparison. For example, how do we know when the system is performing well? How do we know when each module (individual algorithm) within the system is performing well? The connection between the measurements and objectives must be clear and very broadly understood by citizens. Without this understanding, the informed debate demanded by our governance system, and the informed consent of the governed, is unlikely.

Testing

Currently, laws proposed by the United States Congress undergo simulation testing by the Office of Budget Management, and often regulations are subject to simple cost-benefit and environmental evaluation. Helpful as this testing may be, it is inadequate if we are to build responsive and adaptive algorithmic legal systems. More seriously, there is almost no tradition of testing new legal algorithms (whether executed by human bureaucracies or by computers) on a representative (and consenting) sample of communities. This failure to test is hubris, tantamount to believing that we can build systems that are perfect *ab initio*. It is a recipe for creating low-quality legal systems.

Robust adaptive system design

The system of legal algorithms (e.g., a law and associated regulations) must be modular and continuously auditable, with a clear connection between measurement criteria and system goals, such that it is easy to revise or update modules (legal algorithms) and module organization. A failure to implement modern system design tools makes it likelier that the resulting legal system will be opaque, unresponsive to harms, and difficult to update.

Continuous auditing

Systems of legal algorithms (e.g., a a law and associated regulations) must have an operational mechanism for continuous auditing of all modules and overall system performance. Such auditing requires involvement and oversight by all human stakeholders, and must include, by default, the capacity of those stakeholders to modify algorithms or system architecture so that the system meets specified performance goals. The failure to audit ensures that we will have serious failures of our legal system as society and our environment evolve. I suggest that ability to modify algorithms be accomplished by requiring regulators, legislators, and courts (as appropriate) to respond promptly to stakeholder concerns.

Implications for the Practice of Law

What does this mean for lawyers and legislators? Historically, legal careers have begun with the drudgery of wordsmithing and searching through legal documents. In the manner as happened with spell check and web search, this work is now being streamlined by AI-driven document software which searches large document stores to find relevant clauses and suggest common wordings.

These trends are often seen as reducing the demand for legal services, but there are also new opportunities for developing legal agreements using tools originally intended for creating large software systems. These tools are beginning to allow lawyers and legislators to design much more agile, interpretable, and robust legal agreements.

As a consequence, the legal profession has the opportunity to transition from being a cost center and a source of friction, to a center for new business and opportunity creation. The goal of this Computational Law Report is to help seize this opportunity, to support new legal scholars in their enthusiasm for using new digital technologies, and to improve our systems of contracts and governance.

Want to learn about Sandy's related work? His current projects include the Atlas of Inequality, Data Pop Alliance, Open Algorithms, law.MIT.edu, and MIT Connection Science.

Notable profiles of Sandy's research can be found in Harvard Business Review, the UN's Data Revolution Working Group, and by MIT's Sloan School of Management.