Forensic Algorithms: When are (or are not) they appropriate

CSAFE Meeting

June 14, 2021

Karen Kafadar, Panelist

kkafadar@virginia.edu

Brief comments:

1. Algorithms today
2. Conditions encouraging use of algorithms
3. Conditions urging caution in algorithms
4. Validation: Appropriate performance metrics
5. Final thoughts

Thanks to Henry Swofford's webinar:
`forensicstats.org/blog/portfolio/algorithms-in-forensic-science/`
and to Jordan Rodu: `arxiv.org/abs/2001.07648`

1. Algorithms today

1. Everywhere
2. Marketing: targeting customers for purchases
3. Personalized Medicine: Recommended Rx, Tx, procedures (based on person's profile, risk factors, test results)
4. Interest rates: economic conditions
5. Drug development: which combination of compounds
6. Autonomous vehicles
7. **Forensics**: ID vs no-ID (DNA), sentencing, ...

When are/are not algorithms appropriate in forensic science?

Two examples

1. DNA: Algorithms to calculate LR = *likelihood* of suspect
Given the evidence (electropherogram), LR = probability that suspect's DNA *is* (relative to *is not*) "consistent" with evidence.

Notes:

- Electropherogram peaks & locations are measured
- All measurements are measured *with error*
- We hope data (electropherogram) going into an algorithm are measured with "negligle" error
- "Negligible"? Even accounting for error, conclusion is same
- Different algorithms ⇒ different LRs

Result of DNA analysis rarely questioned, unless lab error ⇒ repeat test

2. Algorithmic development underway: Quality Metric

- **Tool** to assist practitioner in inital step of forensic evidence examination: Is the evidence worth time & resources to proceed with further analysis?
- How to define "worth"?
- "high [low] quality" worth [not worth] further analysis
- Metric of "worth": Does level of quality in the evidence indicate high [low] probability of "correct assessment"?
- How is "correct assessment" determined?
  "Ground truth" (known)? "3 agree" (consensus)?
- **QM renders no final decision**; only a tool for deciding whether 'to proceed or not to proceed' with further analysis

Proposal: Presumably the point of analyzing forensic evidence is to identify or exclude an individual as the source of the evidence. Accordingly, LPA-I uses as the "measure":

Probability of correct assessment

(proportion of examiners who correctly include/exclude).

That is, "QM score" can be correlated with how well it predicts its value for law enforcement (correct assessments).

Are there other "measures"?

2. Conditions encouraging use of algorithms

When 'black box' algorithms are (not) appropriate:
A principled prediction-problem ontology
Jordan Rodu, Michael Baiocchi, `arXiv:2001.07648v4`

When can an algorithm be deployed in real-world situations?

"*[E]xtraordinary black-box algorithms exist - with so much potential to do good - despite deep uncertainty about when and how to use them.... [F]or all of their achievements, black-box algorithms have shown [themselves] to be unpredictably brittle in the real world, a consequence of how they are developed.*"

Major challenge: "black-box": We don't understand them.

"*by understanding how they are being developed and assessed, we can understand what situations are more - and less - compatible or safe for their use.*"

Rodu & Baicchi's specific example: *criminal sentencing*

Rodu & Baicchi's example suggests conditions for use of algorithms

1. *Clarity* of *assumptions* used to develop algorithm
2. "*Fairness*" (representativeness) of units in the population to whom [which] algorithm will be applied
   (ex: bicycle alert sign on a highway)
3. *Assessment*: How is algorithm judged to deliver correct answer? (cf. "Performance Metric" question for QM)
4. *Adaptibility*: How does algorithm respond to a *change* in the population for which it was developed?
5. *Costs* of incorrect conclusions/decisions
6. *Justification* for conclusions are clear to users

For their specific example of *Criminal sentencing*:

- *Motivation* for crime may matter.
- Does algorithm consider *motivation*?
- If so, how, and why, is such consideration justified?

3. Conditions urging caution in algorithms

1. *Lack of clarity* in *assumptions* used to develop algorithm
2. Algorithm finds its way into use on populations for which it was neither developed nor tested
3. *Inappropriate assessment* (wrong metric)
   (e.g., "two 'experts' agree": they *both* could be wrong, with small probability multiplied by millions of uses)
4. *Non-adaptibility*: Algorithm is static
5. *Unspecified costs* of incorrect conclusions/decisions
6. *Justification* for conclusions are not clear to users

4. Validation: Appropriate performance metrics

Rodu & Baiocchi argue that the traditional "Common Task Framework" (CTF) for developing algorithms is inadequate:

"*CTF provides an efficient environment for development. The data exist already. All analysts have access to these common data so many people can work on the problem at the same time. Fast computation takes the place of proving theorems, performance is quickly assessed using held-out data, [minimal] consequences of a poorly performing prediction algorithm; e.g., after a failure the analyst tweaks the algorithm.* ***Fundamentally, CTF takes complex real-world problems and sand-boxes them.***

5. Final thoughts

(with thanks to HS & R&B)

We *can* do better with algorithms:

1. "*Start with stakeholders*": who defines the prediction problem, who is *accountable* for its success/failure once deployed, who is affected by its results?
2. Craft the *problem* first, **not** the prediction task
3. Focus on features of the *problem*, **not** on features of the *algorithm*
4. *Transparency* in all phases: assumptions, samples from populations, code

Thank you!