Introduction to Decision Making Algorithms

Jeanna Matthews
Clarkson University
Decision Making Algorithms

• Big decisions about the lives of individuals are being made in a partnership between human decision-makers and computer systems.

• Fundamentally changing the landscape of our societal decision-making processes
  • Criminal justice, hiring, housing, credit, news amplification, elections, ...

• In an environment dominated by trade secrecy, what will be the incentives for iterative improvement/debugging? Fairness? Respect of fundamental societal principles?
Algorithm

• Unambiguous specification of how to accomplish a task
• Step-by-step instructions
• Recipe
How specified?

• Unambiguous specification such as ....
  • Set of written instructions
  • Flow chart
  • Statistical model
  • Program
• No longer ad-hoc and unspecified
How implemented?

• How is algorithm implemented or executed?
  • Humans follow directions?
  • Software? Hardware? Partnerships?

• The more complex the algorithm, the more software or hardware in needed to implement it
  • Automated System
  • Automated Decision Making
Where did the specification come from?

• System designer/developer

• Rule-based systems learned from domain experts

• Learned from data
  • Looking for patterns in data/ “facts” about the world
  • Often still fundamentally learned from humans: manual classification of training data or past data that reflects human decisions
Machine Learning Systems

• Typical process of classification
  • Manual labeling
  • Learning from past “successes”
• Impact of training data
• Dogs in the snow
• Learn from but be careful not to reproduce the past

Defining correctness

• How is correctness defined?
  • Does the implementation faithfully follow the specification?
  • What if the specification is incorrect/incomplete?

• Other metrics?
  • Accuracy of prediction?
  • Impact on society?
Correctness?

• Is it correct?
  • For a particular case?
  • For all cases?
• Are people capable of even determining whether it is correct?
  • Which people?
  • Systems that are too complex to be manually verified.
• Is it understandable/explainable?
  • To which people?
Bugs

• Complex systems and automated systems have bugs
  • Anyone who or uses them knows this!

• They cannot be correct without transparency and iterative improvement
Legal Protections

• Intellectual property claims used to keep away legitimate concerns about correctness

• DeWitt clauses in terms of service documents used to stifle reporting of problems

• Anti-reverse engineering used to prevent thorough third-party testing
Incentive for debugging?

- In this environment, essential to ask “what is the incentive for debugging and iterative improvement?”

- Doomed to run society on buggy systems if we don’t enable iterative improvement
Interests of developers vs. deciders vs. those decided about

• Interests of system developers or system customers are often different than interests of those being decided about
  • Rare cases that matter to individuals
  • Often boils down to efficiency or reduced risk for the decision maker versus protection for the individual
  • Invest some of savings in robust investigation of errors
  • Tax on deciders – but that is not new!

• Criminal justice applications perfect example
  • Interests of developers? Interests of deciders?
  • Rights of defendants? Rights of society?
  • Debugging left to individual defense teams
  • What might be changed out from under us in the process of careless automation without incentives for transparency and iterative improvement
Algorithm = specification

- Specification makes decisions auditable and questionable

- What good is specification if we lock it up in a black-box automated system and don’t allow auditing, questioning?
• What types of review might attorneys and judges seek in understanding software-based/computer-based evidence?

• Why law and public policy require disclosure of these materials to the public and independent experts?
## Executables

<table>
<thead>
<tr>
<th>Locus</th>
<th>Rep1</th>
<th>Rep2</th>
<th>Rep3</th>
</tr>
</thead>
<tbody>
<tr>
<td>D10S1248</td>
<td>13 14 16</td>
<td>13 14 15 16</td>
<td>13 14 15 16</td>
</tr>
<tr>
<td>VWA</td>
<td>15 17 18 19</td>
<td>15 16 17 18 19</td>
<td>15 16 17 18 19</td>
</tr>
<tr>
<td>D16539</td>
<td>11 13</td>
<td>10 11 12 13</td>
<td>10 11 12 13</td>
</tr>
<tr>
<td>D2S1338</td>
<td>17 18 19</td>
<td>17 18</td>
<td>17 18 19</td>
</tr>
<tr>
<td>D8S1170</td>
<td>10 12 14 15</td>
<td>10 12 14 15</td>
<td>10 11 12 14 15</td>
</tr>
<tr>
<td>D2 1511</td>
<td>28 29 30 32.2</td>
<td>28 29 30 32.2</td>
<td>28 30 32.2</td>
</tr>
<tr>
<td>D1851</td>
<td>14 15 17</td>
<td>10 14 15 17</td>
<td>14 15 18</td>
</tr>
<tr>
<td>D2 751045</td>
<td>11 12 15 16</td>
<td>12 15 16</td>
<td>11 12 15 16</td>
</tr>
<tr>
<td>D3 9413</td>
<td>13 14</td>
<td>13 14</td>
<td>13 14</td>
</tr>
<tr>
<td>Th01</td>
<td>6 7 8 9.3</td>
<td>7 8 9 3</td>
<td>7 8 0.3</td>
</tr>
<tr>
<td>FGA</td>
<td>18 21 24</td>
<td>18 19 22 23 24</td>
<td>18 21 22 23 24</td>
</tr>
<tr>
<td>D25441</td>
<td>10 11</td>
<td>10</td>
<td>10 11 11.3</td>
</tr>
<tr>
<td>D151358</td>
<td>16 17</td>
<td>15 16 17</td>
<td>15 16</td>
</tr>
<tr>
<td>D161656</td>
<td>12 15 16 18</td>
<td>13 12 15 16</td>
<td>12 16</td>
</tr>
<tr>
<td>D1 25391</td>
<td>15 17 19 20 23</td>
<td>15 17 19 20</td>
<td>15 17 19 20</td>
</tr>
</tbody>
</table>
public void CheckFrequencyForRemoval(DataTable dtFrequencies)
{
    // If our db connection isn’t initialized, do it first, then, get all the ethnicities (races)
    MySqlConnection mydb = mydb ?? new MySqlConnection();
    DataTable raceTable = mydb.GetTable("Ethnicities");
    List<int> r = new List<int>(raceTable.Rows.Count);
    foreach (DataRow row in raceTable.Rows)
    {
        r.Add((int)row[0]);
    }

    // We go through all the comparison loci and check whether the sum of the frequencies for that locus is greater than 0.97.
    // If it is, we remove the locus. Frequencies are only used for the alleles in the evidence replicates.
    for (int i = 0; i < comparisonLocici.Count; i++)
    {
        bool bRemove = false;
        // Get a csv list of alleles for all the replicates at a locus
        string unknownPair = evidenceAlleles[(unknownAlleles[comparisonLocici[i]])];
        // check if the frequency is greater than 0.97 for any of the races. Frequencies are values for an allele at a locus
        foreach (DataRow eachRow in raceTable.Rows)
        {
            string raceName = eachRow.Field<string>("EthnicName");
            float frequency = GetFrequencyValues(unknownPair, comparisonLocici[i], raceName, dtFrequencies);

            if (frequency >= 0.97)
            {
                bRemove = true;
                break;
            }
        }
        if (bRemove)
        {
            sre[i] = comparisonLocici[i];
        }
    }
}
Other parts of specification

• Information from the development process
  • Design documents, testing plans and results

• Experience with deployed software
  • Bug reports, change logs
Brown Institute Magic Grant: Decoding Differences in Forensic DNA Software
Methods

• Independent, third-party, adversarial testing and review
  • Automated testing harnesses
  • Common file formats and settings
  • Source code analysis

• Recommendations
  • Clear advise for judges, defense attorneys, journalists
  • Sample requirements for software systems, targeting the procurement phase
DEFCON Talk: “You’re Just Complaining Because You’re Guilty”
•Upcoming article in AI Magazine and talk at 2019 AI For Good

“Patterns and Anti-Patterns, Principles and Pitfalls: Accountability and Transparency in AI“

•10 Common Anti-Patterns

1. Learn from the Past Without Remembering the Context
2. Learning from Humans Without Remembering Human Bias and the Possibility of Malicious Training
3. Using Data You Have Rather than the Data You Need
4. Failing to Measure the Social Impact of Deployed Systems
5. ....
Final Words

• Introduction to decision making algorithms

• Human decision making vs. automated decision making
  • Specificity, Repeatability, Complexity

• Importance of Incentivizing Iterative Improvement

• Protection for Individuals and the Public Good (Not Just Efficiency and Reduced Risk for Deciders)
Thank you!

jnm@clarkson.edu

http://www.clarkson.edu/~jnm

@jeanna_matthews