Logical Induction: a computable approach to logical non-omniscience

Adrià Garriga Alonso

February 21, 2018

Why?

Reasoning about logical implications takes computation time. Some mathematical questions have no proof (yet?), but they have evidence in favour.

- $P \neq NP$
- Twin prime conjecture: “There are infinitely many pairs of consecutive primes that differ by 2 units” (e.g. 5 and 7, 41 and 43)
- Riemann Hypothesis

Computing more facts, without observing new evidence, should change our beliefs.

- Proof by Zhang [2014] that prime gap is $< 7 \cdot 10^7$ should increase confidence in the twin prime conjecture.
The Dutch Book argument

Let $S$ be sentences in some logical language $\mathcal{L}$. The agent’s belief $\mathbb{P} : S \mapsto [0, 1]$ for proposition $\phi \in S$ is its fair price for a bet $\mathbb{W} : S \mapsto \{0, 1\}$ that pays:

$$
\mathbb{W}(\phi) = \begin{cases} 
1 & \text{if } \phi \text{ is true} \\
0 & \text{if } \phi \text{ is false}
\end{cases}
$$

If the belief does not satisfy axioms of probability theory then there is a set of trades on bets (a “Dutch Book”) that guarantees the agent loses money.

<table>
<thead>
<tr>
<th>Proposition $\phi$</th>
<th>Agent’s belief $\mathbb{P}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>heads</td>
<td>0.6</td>
</tr>
<tr>
<td>$\neg$heads</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Dutch book: sell 1 unit of “heads” and one of “$\neg$heads”.

The Dutch Book for logical sentences

Fix a logical language $\mathcal{L}$ that includes propositional logic (the usual $\neg$, $\land$, $\lor$, $\rightarrow$, ...), and a set of axioms $\Gamma$ in that $\mathcal{L}$. The only inference rule needed is *modus ponens*, if by a sequence of *modus ponens* we get theorem $\phi$ then we write $\Gamma \vdash \phi$. Now:

$$W(\phi) = \begin{cases} 
1 & \text{if } \Gamma \vdash \phi \\
0 & \text{if } \Gamma \vdash \neg \phi \\
\text{undefined for now} & \text{if } \Gamma \not\vdash \phi \text{ and } \Gamma \not\vdash \neg \phi
\end{cases}$$

For a given $\Gamma$ like Peano Arithmetic (PA) or ZFC:

<table>
<thead>
<tr>
<th>Proposition $\phi$</th>
<th>Agent’s belief $\mathbb{P}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2 + 2 = 4$</td>
<td>1.0</td>
</tr>
<tr>
<td>$4 - 1 = 3$</td>
<td>0.99</td>
</tr>
<tr>
<td>The 613487th digit of $\pi$ is a 6</td>
<td>0.11</td>
</tr>
<tr>
<td>$P \neq NP$</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Dutch Book: buy a bet for all provable $\phi$. 
Relaxed criterion for a computable reasoner

**Problem**: deciding whether $\Gamma \vdash \phi$ is uncomputable! (read: requires infinite computation time, as opposed to arbitrarily large but finite).

No “Dutch Book” exists that can extract money with probability 1 from the agent’s beliefs.

\[\Downarrow\]

No “Dutch Book” **that can be written in polynomial time** exists [...].
Polynomial in what?

In the number of days!

- We’ll construct a market in which shares for logical sentences can be bought and sold by traders.

- Every day $n$, a deductive process $\bar{D} := D_1, D_2, \ldots$ marks some sentences as theorems. The shares for those sentences will (roughly) pay £1 to traders which bought them, for every future day $m \geq n$.

- An algorithm for computing a market is a logical inductor if no efficiently computable trader can exploit it.
  - A trader $\bar{T}$ is efficiently computable if the number of steps it takes to compute its trading strategy (Dutch Book) for day $n$ is polynomial in $n$ (note: not $\log_2 n$).
A **deductive process** $\overline{D}$ is a computable nested sequence $D_1 \subseteq D_2 \subseteq \ldots$ of finite sets of finite sentences. We write $D_\infty$ for the union $\bigcup_n D_n$.

- We say $\overline{D}$ is $\Gamma$-complete if $\phi \in D_\infty \iff \Gamma \vdash \phi$.
- For example, when $D_n$ is the set of sentences $\phi \in \mathcal{S}$ that can be proven from $\Gamma$ in at most $n$ characters.

A **market** $\overline{P} := P_1, P_2, \ldots$ is a computable sequence of pricings (beliefs): $P_i : \mathcal{S} \mapsto [0, 1] \cap \mathbb{Q}$.

An **expressible feature** $\xi$ of rank $n$ is a *continuous* function from pricing sequences to real numbers, such that it only depends on the first $n$ pricings. Informally, $\xi(n, \overline{P}) = \xi(n, P_{\leq n}) = v$. 
Formal definitions: Traders and strategies

- A **trader** is a sequence $\overline{T} := T_1, T_2, \ldots$ of trading strategies (Dutch Books) for day $n$.

- A **trading strategy** $T_n$ gives the number of shares of each $\phi$ to buy/sell as a linear combination of sentences, weighted by features. That is:

  $$T_n := \sum_{i}^{M} \xi_i(n, P_{\leq n}) \cdot \phi_i - c$$

  - Trading strategies can depend on the current prices.
  - $c = \sum_{i} \xi_i \phi_i^{*n}$ is a constant that reflects the cash paid by the trader in day $n$. The $\phi_i^{*n}$ are features that represent the price of $\phi$ at time $n$.
  - Negative prices or cash represent selling.
How do the bets pay? Consistent worlds.

- A world $\mathcal{W}: \mathcal{S} \mapsto \{0, 1\}$ is **propositionally consistent** with a set of sentences $D$ if $\mathcal{W}(\phi)$ is consistent with the truth values of prime sentences of $\phi$. So: $\mathcal{W}(\phi \land \psi) = \mathcal{W}(\phi) \land \mathcal{W}(\psi)$, . . .
  - Write $\mathcal{P}C(D)$ for the set of worlds p.c. with $D$.
  - Write $\mathcal{P}C(\Gamma)$ for worlds consistent with $\Gamma$, that is

\[
\Gamma \cup \{\phi \mid \mathcal{W}(\phi) = 1\} \cup \{\neg \phi \mid \mathcal{W}(\phi) = 0\} \not\models \bot.
\]
The Logical Induction Criterion

- A trader $\overline{T}$ is said to **exploit** a market $\overline{P}$ relative to a deductive process $\overline{D}$ if the set of values

$$\left\{ \mathbb{W}\left(\sum_{i\leq n} T_i(\overline{P})\right) \mid n \in \mathbb{N}^+, \mathbb{W} \in \mathcal{PC}(D_n) \right\}$$

is bounded below, but not bounded above.

  - Note: traders can exploit the market even if they never purchase a provable sentence. Suppose $\text{PA} \vdash (\phi \vee \psi)$ but $\text{PA} \nvdash \phi$ and $\text{PA} \nvdash \psi$. Then if $\phi$ and $\psi$ were purchased for less than £1, the trade is positive.

- A market $\overline{P}$ is said to satisfy the **logical induction criterion** relative to a deductive process $\overline{D}$ if there is no efficiently computable trader $\overline{T}$ that exploits $\overline{P}$ relative to $\overline{D}$. 
Desirable properties

Unfortunately, all of these only hold in the limit, that is, when $n \to \infty$.

1. **Convergence and Coherence**: when $n \to \infty$, the prices reflect a logically consistent belief state, and a probability distribution over all possible worlds.

2. **Timely Learning**: any e.c. sequence of theorems is identified as such relatively fast, regardless proof length.

3. **Learning Statistical Patterns**: e.g. 10% probability to the $n$th digit of $\pi$.

4. **Learning Logical Relationships**: probabilities consistent with propositional logic operators.

5. **Non-Dogmatism**: only assign 1 or 0 to provable sentences.
Desirable properties 2

7. **Conditionals**: Conditioning a LI on an extra logical sentence (dividing $\overline{P}$s) gives another LI.
   - **Recovers Bayesianism**: if conditioned on a logically independent sentence.

8. **Trust in Consistency** of its own, and stronger, theories $\Gamma$, if they are in fact consistent. Note that when $n \to \infty$ the probability they assign to consistency is less than 1, thus not breaking Gödel’s incompleteness theorem.

9. **Reasoning about Halting**: if there’s an efficient method for generating programs that halt.

10. **Introspection**: Logical inductors “know what they know”, in that their beliefs about their current probabilities and expectations are accurate.

11. **Self-Trust**: Logical inductors trust their future beliefs.
Undesirable properties

1. **No meta-cognition**: Li cannot decide which sentences to think about.
   1.1 Don’t have goals, thus no reason to prefer one sentence over the other.
   1.2 Given a sentence to reason about, the nice properties when $n \to \infty$ might not apply.

2. **No answers to counterpossible questions**: e.g. in 1993, ask “what would happen if Fermat’s Last Theorem is false”? \(\perp\) implies everything, so no meaningful answer even in the limit.

3. **Uncomputable convergence rate**.

Algorithm

$LIA_{\leq 0} = ()$

$LIA_n = \text{MarketMaker}_n(\text{TradingFirm}_n^{\bar{D}}(LIA_{\leq n-1}), LIA_{\leq n-1})$

$\text{MarketMaker}(\tau_n, \mathbb{P}_{\leq n-1})$: set the prices to the trader $\tau_n$'s “fair prices”, such that they abstain from betting, except to buy shares very close to £0 or sell them close to £1.

$\text{TradingFirm}_n^{\bar{D}}(\leq n-1) = \sum_{k \in \mathbb{N}^+} \sum_{b \in \mathbb{N}^+} 2^{-k-b} \cdot \text{Budgeter}_n^{\bar{D}}(b, S^k_n, \mathbb{P}_{\leq n-1})$

$\bar{T}^k$: enumeration of all possible efficiently computable traders.

$S^k_n = \begin{cases} 
T_n^k & \text{if } n \geq k \\
0 & \text{otherwise}
\end{cases}$
Closing thoughts

- Similar to Solomonoff induction: uncomputable (in practice) models for idealised induction.
  - First step towards modeling agents that reason about themselves with bounded computational resources.
- Likely already usable as a replacement in models assuming logically omniscient / perfectly Bayesian reasoners.
- Would be nice to have it perform in practice, when not enumerating all possible efficiently computable traders.