Logical Induction
Abridged version, early draft

Scott Garrabrant, Tsvi Benson-Tilsen, Andrew Critch, Nate Soares and **Jessica Taylor**
{scott,tsvi,critch,nate,jessica}@intelligence.org

August 6, 2016

Abstract

We present a computable algorithm for assigning probabilities to sentences of logic, such as sentences of the form "this long-running computation outputs 3" or "the twin prime conjecture is true". This algorithm can be seen as an inductive process that out-paces deduction, in that it learns to accurately predict the results of long-running deductions well before they finish, by observation of shorter-running ones. The algorithm has a number of good inductive, deductive, and reflective properties. It inductively learns to assign probabilities that respect logical relationships between sentences (such as that a given computer program only ever produces a certain type of output) in a timely manner, so long as the proposed relationship can be written down in polynomial time. It inductively learns to fall back on appropriate statistical summaries in the face of sequences that appear pseudorandom (such as "the nth digit of π is a 7" for large n). In the limit it obeys all logical constraints and learns all logical facts; our algorithm can in fact be viewed as a computable approximation to a probability distribution that dominates the universal semimeasure. It trusts its own probabilities, in the sense that, roughly speaking, it knows what it knows, and trusts its future beliefs to be more accurate than its current beliefs.

These properties and others all follow from a single *logical induction criterion*, which is motivated by a series of stock trading analogies. Roughly speaking, we interpret the belief-state of a logically uncertain reasoner as a set of market prices: $P(\phi) = 50\%$ is interpreted as saying that a note that pays out \$1 if ϕ is proven costs 50 cents. The logical induction criterion states that it should not be possible for a stock trader to exploit a good logical reasoner's prices, e.g., via arbitrage, using any strategy that can be implemented in polynomial time. This criterion bears strong resemblance to the "no Dutch book" criteria that support both expected utility theory (von Neumann and Morgenstern 1944) and Bayesian probability theory (Ramsey 1931; de Finetti 1979).

1 Introduction

Consider encountering a computer connected to an input wire and an output wire. If you know what computer program it implements, then there are two distinct ways to be uncertain about its outputs. You could be uncertain about the input—maybe it's determined by a coin toss you didn't see. Alternatively, you could know the input, but be uncertain because you haven't had the time to reason out what the program actually outputs—perhaps the program searches efficiently for proofs of the Riemann hypothesis for a large finite amount of time, and you know exactly how the program works, but you're not sure whether the search will succeed.

The first type of uncertainty is about empirical facts. No amount of thinking about the coin toss will tell you its value until you observe the result. When you

Research supported by the Machine Intelligence Research Institute (intelligence.org).

make new observations that provide evidence about empirical facts, probability theory gives a principled account of how to manage that kind of uncertainty, via Bayes' theorem.

The second type of uncertainty is about *logical* facts, such as how a known program actually behaves. In this case, reasoning alone in a dark room *can* change your beliefs over time. You can reduce your uncertainty by thinking more about the program, without making any new observations of the external world.

It's not yet clear, in principle, how to manage logical uncertainty, nor what it means for a reasoner to do "good reasoning" about logical facts in the face of deductive limitations. For example, every mathematician has experienced uncertainty about conjectures for which there is "quite a bit of evidence", such as the Riemann hypothesis or the Goldbach conjecture. When the authors heard that Ramaré (1995) had showed that every even number is the sum of at most 6 primes, we were tempted to increase our credence in the Goldbach conjecture. How much evidence does Ramare's proof provide? Can we quantify the degree to which it should increase our confidence?

The natural impulse is to appeal to probability theory in general and Bayes' theorem in particular. Bayes' theorem gives rules for how to use observations to manage empirical uncertainty about unknown events in the physical world. However, uncertainty about a deterministic proposition like $\phi =$ "the 10^{100}th digit of π is 7" is of a different nature. No amount of reasoning about a fair coin flip will change one's probability from 50% on either outcome until the coin is revealed. But thinking about ϕ for a very long time, you could (rightly) move your credence in ϕ from something agnostic (like 10%) to something close to 0% or 100%, without ever observing a new piece of sensory data.

Probability theory lacks the tools to describe this sort of "logical update". For example, suppose the 10^{100}th digit of π is 8, in which case ϕ is false. Then we have the implication $\top \to \neg\phi$, where the left hand side is tautological, and so has probability 1. But in probability theory, when $A \to B$, we must have $\mathbb{P}(A) \leq \mathbb{P}(B)$. Hence, a probability-theoretic reasoner must assign probability ≈ 1 to $\neg\phi$!

Many myriad solutions have been proposed for solving this problem. For example, Hacking (1967), Demski (2012), and Christiano (2014) propose different methods for relaxing the restraint that $P(A) \leq P(B)$ until such a time as the implication $A \to B$ has been proven. But then there remains the question of how probabilities should be assigned before an implication is proven (such as when an implication is suspected), and this brings us back to the search for a principled method for assigning probabilities to unproven mathematical conjectures, and updating them in response to the discovery of new mathematical facts. It is this question that logical induction attempts to address.

We propose a solution, which we call the *logical induction criterion*. Our basic setup works as follows. We will consider reasoners that assign explicit probabilities to sentences of logic and refine those probabilities over time, where those sentences can make claims like "`prg(i)`=0" or "the 10^{100}th digit of π is 7" or "Goldbach's conjecture is true". We will interpret a reasoner's probabilities as prices in a stock market, where the probability that a reasoner assigns to a sentence ϕ is interpreted as the price of a note that pays out \$1 if ϕ turns out to be true. We will have new knowledge of logical facts enter the market each day, via some slow deductive process (such as a theorem prover). We will consider a collection of efficient (polynomial-time) stock traders who buy and sell shares at the market prices using trading strategies that vary continuously in the market prices. (The reason for the continuity constraint will be discussed later; in short, it allows the market to avoid the classic paradoxes of self-reference.) Our criterion then says that the market prices that a good reasoner writes down should not be exploitable, in the sense that no stock trader should be able to attain unbounded returns off of a finite risk.

The logical induction criterion can be seen as a weakening of the "no Dutch book" criterion that (Ramsey 1931; de Finetti 1979) used to support standard probability theory. Under this interpretation, our property says (roughly) that a rational deductively limited reasoner should have beliefs that can't be exploited indefinitely by any Dutch book constructed by an efficient (polynomial-time) algorithm.

Logical inductors satisfy many desirable properties, including:

- They are logically consistent in the limit.
- They dominate the universal semimeasure in the limit.
- They learn inductively to assign high probability to any pattern of theorems that can be generated in polynomial-time, even if the underlying deductive process is very slow to prove those theorems.
- They learn to make their probabilities of statements respect linear inequalities that will hold of the truth values of those statements in the limit.
- They learn to use appropriate statistical summaries on apparently pseudorandom sequences (such as "the nth digit of π is a 7" for large n).
- They learn to correctly answer questions about their own beliefs (such as "is your probability of ϕ between a and b?") while avoiding paradox.
- They learn to trust their future beliefs.

Below, we will formalize precise versions of each of these claims.

Our main result is a computable algorithm which implements a logical inductor. This paper is an abbreviated version of a longer paper, and so we will only prove a handful of the above claims, after giving the main algorithm. The extended paper contains proofs of the rest, as well as proofs of generalized and strengthened versions of the properties listed here, as well as proofs of a number of other properties (calibration, learning to trust the underlying theory, the behavior of conditional probabilities, etc.). The extended version also includes speculation about why this framework works, and a discussion of potential applications. It has not yet been published; email `rob@intelligence.org` if you would like a copy upon its release.

1.1 Related work

The body of related work is too large to do justice here. Our algorithm follows in the footsteps of Gaifman (1964), who considers the problem of assigning coherent probabilities to sentences of logic. This approach has also been pursued by Demski (2012), Christiano (2014), and Hutter et al. (2013), among others. We draw heavy inspiration from Solomonoff's theory of inductive inference (Solomonoff 1964) and the related theoretical work of L.A. Levin, e.g., on the universal semimeasure (Levin 1974). We also draw heavy inspiration from the "no Dutch book" criteria that support probability theory (Ramsey 1931; de Finetti 1979) and expected utility theory (von Neumann and Morgenstern 1944).

Reasoning under logical uncertainty has been studied by many over the years, in many different contexts. For a discussion of the problem of "logical omniscience" (many natural models of good reasoning implicitly require the reasoner to be omniscient about logical facts), refer to Hintikka (1975), Rantala (1975), and Fagin et al. (1995). Refer to Good (1950) and Hacking (1967) for a discussion of ways to weaken probability theory to allow for deductive limitations. Refer to Carnap (1962) and Savage (1967, Sec. 53) for some discussions of the different desiderata that logically uncertain reasoners should satisfy. Refer to Hintikka (1962) and Campbell-Moore (2015) for a discussion of the difficulties that arise when attempting to define reasoners that know what they know. In the field of AI, refer to Hay et al. (2012) for a discussion of the problem of reasoning about what to reason about under deductive limitations. For a discussion of the relationships between logical uncertainty and computational complexity theory, refer to Aaronson (2013). This list is woefully incomplete; refer to Hutter et al. (2013) for a more thorough review of the study of logical uncertainty across the years, and refer to the extended version of this paper for a more thorough discussion of how different works relate to our results in particular.

2 Setup

2.1 Notation

We will put logical sentences in quotes, like $\phi :=$ "$x = 3$". We will use underlines to denote the splicing of a metalanguage variable into a logical sentence, for example, if $\psi :=$ "$y > 3 \to \underline{\phi}$" then ψ denotes the sentence "$y > 3 \to x = 3$". If ϕ and ψ denote sentences, then $\neg\phi$ denotes "$\neg(\underline{\phi})$" and $\phi \wedge \psi$ denotes "$(\underline{\phi}) \wedge (\underline{\psi})$" and so on. If f is some computable function, it is understood that "$\underline{f}(7)$" can be expanded into a precise claim written in the language of Peano Arithmetic (PA). Similarly, we will sometimes write things like $\phi :=$ "Goldbach's conjecture", in which case it is understood that the English text could be expanded into a precise arithmetical claim.

We define $\mathbb{N}^+ = (1, 2, \ldots)$ to be the positive natural numbers. We use $\mathbb{N}^+$ instead of $\mathbb{N}^{\geq 0}$ because we will commonly consider initial segments of infinite sequences up to and including the element at index n, and it will be convenient for those lists to have length n. When we discuss numbers colloquially (such as saying "on each day n"), it should be understood that we are implicitly quantifying over $\mathbb{N}^+$

We denote infinite sequences using boldface, like $\boldsymbol{x} = (x_1, x_2, \ldots)$, where it is understood that x_i denotes the ith element of $\boldsymbol{x}$, for $i \in \mathbb{N}^+$. Boldface letters usually denote finite lists; we will find it convenient to use them to denote infinite lists, as we regularly manipulate infinite sequences. When writing sequences out, we use parenthesis, like $\boldsymbol{\psi} = (\text{"}\underline{n} > 7\text{"})_{n \in \mathbb{N}^+}$ which defines the sequence ("$1 > 7$", "$2 > 7$", "$3 > 7$", ...).

Given any sequences $\boldsymbol{x}$ and $\boldsymbol{y}$, we write

$$\begin{aligned} x_n &\eqsim_n y_n \quad \text{for} \quad \lim_{n\to\infty} x_n - y_n = 0, \\ x_n &\gtrsim_n y_n \quad \text{for} \quad \liminf_{n\to\infty} x_n - y_n \geq 0, \text{ and} \\ x_n &\lesssim_n y_n \quad \text{for} \quad \limsup_{n\to\infty} x_n - y_n \leq 0. \end{aligned}$$

2.2 Uncertain reasoners

You might know that "$\top$" is true, and that "$\neg\neg\top$" is true, but do you know whether

$$\text{"}\neg\neg\neg\neg\neg\neg\neg\neg\neg\neg\neg\neg\top\text{"}$$

is true? To find out, you have to pause and apply a pattern. To sidestep the problem of defining whether or not a reasoner "really knows" a pattern, we will consider reasoners that write out explicit beliefs about logical sentences, and refine them over time.

Definition 2.1 (Explicit belief state)**.** *An **explicit belief state** $P : \Lambda \to [0, 1] \cap \mathbb{Q}$ is a computable function from sentences to rational probabilities with finite support.*

Definition 2.2 (Reasoning process)**.** *A **reasoning process** is a computable sequence $\boldsymbol{P} = (P_1, P_2, \ldots)$ of explicit belief states, interpreted as a reasoner's explicit beliefs about logic as they are refined over time.*

You can visualize an explicit belief state as a giant row of cells, one for each logical sentence, where the reasoner has written a probability in finitely many of those cells. You can visualize a reasoning process as a great table of explicit belief states, where the reasoner adds one new one per day. You can visualize the reasoner as a person who writes rational probabilities in the cells, perhaps using computer assistance, as they learn things about logic. For example, once they notice the $\neg^{2n}\top$ pattern, they might program their computer to write 1 in the $\neg^{2n}\top$ cells as fast as possible.

We will be concerned with reasoning processes that contain explicit patterns, such as sequences of theorems that are all given high probabilities in a timely fashion. We mainly consider sequences of sentences that can be generated by some reasoner with polynomial runtime.

Definition 2.3 (Efficiently enumerable)**.** *An infinite sequence $\boldsymbol{\phi}$ is called **efficiently enumerable**, abbreviated e.e., if there is a computable function f that outputs ϕ_n on input n, with runtime polynomial in n.*

When checking whether a reasoning process has begun explicitly "recognizing" an e.e. sequence of sentences, we will check whether the reasoning process has begun assigning specific probabilities to at least one new instance of the pattern per day.

Definition 2.4 (Timely manner)**.** *Let $\boldsymbol{\phi}$ be an e.e. sequence of sentences, and $\boldsymbol{p}$ a sequence of probabilities. We say that a reasoner $\boldsymbol{P}$ assigns $\boldsymbol{p}$ to $\boldsymbol{\phi}$ in a **timely manner** if for every $\varepsilon > 0$, there exists a time N such that for all $n > N$,*

$$|P_n(\phi_n) - p_n| < \varepsilon.$$

In other words, $\boldsymbol{P}$ assigns $\boldsymbol{p}$ to $\boldsymbol{\phi}$ in a timely manner if the timely responses of $\boldsymbol{P}$ to $\boldsymbol{\phi}$ satisfy

$$P_n(\phi_n) \eqsim_n p_n.$$

For example, let ϕ_n be $2n$ $\neg$ symbols followed by a $\top$ symbol. If there comes a day when the reasoner is assigning ϕ_n a probability near 1 no later than the nth day, we will say that they have begun assigning probabilities near 1 to the $\boldsymbol{\phi}$ in a timely manner, and we will sometimes colloquially say that their explicit beliefs have started respecting the pattern. Checking the nth element of the sequence on the nth day is a rather arbitrary choice; the specific diagonal chosen does not matter. For instance, we will consider reasoners that assign *every* e.e. sequence of theorems probabilities near 1 in a timely manner, and because $(\phi_{2n})_{n\in\mathbb{N}^+}$ and $(\phi_{2n+1})_{n\in\mathbb{N}^+}$ are both efficiently enumerable when $\boldsymbol{\phi}$ is, any reasoner with this property must eventually start taking the ϕ_n at a rate of at least two per day (and at least three per day, and so on).

2.3 Markets and traders

We will interpret the probabilities that the reasoner writes down as the prices in a market populated by continuous, polynomial-time traders. We now define markets and traders.

Definition 2.5 (Price table)**.** *A **price table** $P : \Lambda \to [0,1] \cap \mathbb{Q}$ is any computable function from sentences to rational probabilities. We refer to $P(\phi)$ as the price of ϕ, and if $P(\phi) = p$ we say that the price of a ϕ-share is p (where it is understood that a ϕ-share pays out \$1 if ϕ gets proven).*

Definition 2.6 (Market)**.** *A **market** is a computable sequence $\boldsymbol{P} = (P_1, P_2, \ldots)$ of price tables.*

We denote both markets and reasoning processes by $\boldsymbol{P}$, because throughout this paper, we will only ever consider reasoning processes that are also markets. Markets differ in that they do not require finite support at all times. (In other works, we will consider logical inductors that satisfy the logical induction criterion, but do not have finite support.)

Traders will use the following two data types to make their trades:

Definition 2.7 (Market history)**.** *The **market history** of $\boldsymbol{P}$ on day n is the sequence $\boldsymbol{P}_{\leq n} = (P_1, P_2, \ldots, P_n)$ of price tables in $\boldsymbol{P}$ up to and including the prices on day n. We sometimes call $\boldsymbol{P}_{\leq n}$ the **n-history** of $\boldsymbol{P}$.*

Definition 2.8 (Market feature)**.** *A **market feature** $x : [0,1]^{n\times\Lambda} \to \mathbb{R}$ for day n is a continuous rational function[1] from n-histories $\boldsymbol{P}_{\leq n}$ to real numbers $x(\boldsymbol{P}_{\leq n})$. We will commonly assume that $\boldsymbol{P}$ and n are clear from context, and abbreviate $x(\boldsymbol{P}_{\leq n})$ as x, and treat market features as scalars. We write R_n for the set of market n-features.*

1. A market feature can be represented as an expression composed from rational numbers, addition, multiplication, a "safe reciprocation" function $1/\max(\cdot, 1)$, unique symbols for each $P_i(\phi)$, a finite list of variables, and a dictionary that assigns each variable to another expression. The dictionary is necessary to make it possible to write out interesting

Definition 2.9 (Trade)**.** *A **trade** r is a finite linear combination of sentences with real coefficients:*

$$r = \alpha_1\phi_1 + \cdots + \alpha_k\phi_k.$$

We write $r(\phi)$ for the coefficient $\alpha \in \mathbb{R}$ of a sentence ϕ in r. The number $r(\phi)$ will be interpreted as a number of shares, owned or bought, which will settle to \$1 per share if ϕ is determined to be true, or \$0 if ϕ is determined to be false. We measure trades with the ℓ_1 norm, i.e., $|r| := \sum_{i=1}^{k} \alpha_i$. We denote the set of trades by $\mathbb{R}\langle\Lambda\rangle$.[2]

Definition 2.10 (Trading strategy)**.** *A **trading strategy** for day n, also called an **n-strategy**, is a finite linear combination of sentences with market n-features as coefficients:*

$$t = x_1\phi_1 + \cdots + x_k\phi_k.$$

We write: $t(\phi)$ for the coefficient x of ϕ, and $t(\boldsymbol{P}_{\leq n})$ for the trade $\sum_i x_i(\boldsymbol{P}_{\leq n})\phi_i$. Thus t can be viewed either as a finite-support function from sentences to market n-features, or as a function from market n-histories to trades. We denote the set of trading n-strategies by $R_n\langle\Lambda\rangle$.[3]

The reason for the continuity constraint is to allow stable market prices to be found in the face of paradoxes. For example, consider the sentence ϕ which says "the price of ϕ in this market on day n is less than 50 cents" (constructed, e.g., by Gödel's diagonal lemma). What should the fair price be? ϕ pays out \$1 if its price was less than 50 cents, and \$0 otherwise. Thus, if the price of ϕ is low, it is worth a lot; but if the price is high, it is worthless. If traders are allowed to buy ϕ on day n if it costs less than 50 cents and sell otherwise, then there are no stable market prices. The continuity constraint requires that if a trader buys at one price and sells at another, there must be an intermediate price at which they buy/sell nothing. As we will see later, this guarantees that a stable fixed point can be found between market prices and trading strategies.

The continuity condition can be interpreted as saying that traders have only finite-precision access to the market prices.

Definition 2.11 (Trader)**.** *A **trader** is a computable function $\boldsymbol{t} : (n : \mathbb{N}^+) \to R_n\langle\Lambda\rangle$ which takes the day n as input, and produces a trading strategy for day n as output, with runtime polynomial in n. In other words, a trader is an efficiently enumerable sequence of trade strategies. We write t_n for the trading strategy the trader produces on day n, and thus $t_n(\phi)$ for the market feature describing how many shares of ϕ the trader buys or sells on day n (sometimes interpreted as a scalar). We write $\mathcal{T}$ for the set of all traders.*

2.4 How knowledge enters the market

We will imagine ϕ-shares paying out if and when the sentence ϕ is proven. We remain agnostic about exactly how logical facts enter the market, and instead consider markets paired with a sequence of sets of "seemingly plausible" truth assignments that get whittled down over time.

Definition 2.12 (World)**.** *A **world** W is any function $\Lambda \to \mathbb{B}$, i.e., it is a (possibly inconsistent) assignment of truth values to sentences in Λ.*

---

expressions efficiently; without the ability to factor sub-expressions out into variables, some simple patterns of expressions are impossible to generate in polynomial time. This definition is somewhat arbitrary; refer to the extended paper for details, examples, and a discussion of variations. What actually matters is that market features must be (1) smooth in a fashion that allows stable market prices to be found; (2) compactly specifiable in polynomial time; and (3) expressive enough to identify a variety of inefficiencies in a market. The reasons for these requirements will be discussed shortly.

2. We use this notation because the set of all trades is equal to the vector space spanned by Λ over $\mathbb{R}$.

3. We use this notation because the set of all trading n-strategies is equal to the free module spanned by Λ over R_n, which is a commutative ring.

This terminology is nonstandard; the word "world" is usually reserved for consistent truth assignments. But when reasoning under logical uncertainty, reasoners can't tell the difference between contradictory and non-contradictory worlds, so we make a distinction between arbitrary truth assignments ("worlds") and consistent truth assignments.

Definition 2.13 (Consistent world)**.** *A world W is called **consistent** if*

$$\Gamma \cup \{\phi \mid W(\phi) = 1\} \cup \{\neg\phi \mid W(\phi) = 0\}$$

is consistent.

Definition 2.14 (Deductive state)**.** *A **deductive state** D is a non-empty set of worlds.*

Definition 2.15 (Plausible world)**.** *A world W is called **plausible** (relative to a deductive state D) if $W \in D$.*

Definition 2.16 (Deductive process)**.** *Given a sequence of deductive states $\boldsymbol{D}$, define the function* $\text{Plausible}^{\boldsymbol{D}} : \mathbb{N}^+ \times \Lambda^{<\omega} \times \Lambda^{<\omega} \to \mathbb{B}$ *to take two finite lists of sentences $(\phi_1, \ldots, \phi_j)$ and $(\psi_1, \ldots, \psi_k)$, and return 1 if D_n contains a world in which all of the ϕ_i are true and all of the $\psi_i \in \boldsymbol{\psi}$ are false, and 0 otherwise. $\boldsymbol{D}$ is called a **deductive process** if*

1. *$\boldsymbol{D}$ is decreasing, i.e., if $j > i \Rightarrow D_j \subseteq D_i$,*
2. $\text{Plausible}^{\boldsymbol{D}}$ *is computable.*

Definition 2.17 (Then-plausible worlds)**.** *Given a deductive process $\boldsymbol{D}$, the set $\mathcal{W}$ of **then-plausible worlds** is the collection of pairs (W, n) of worlds W that were plausible on some day according to $\boldsymbol{D}$, paired with days n on which they were plausible:*

$$\mathcal{W} := \bigsqcup_n D_n = \bigcup_n \bigcup_{W \in D_n} (W, n).$$

We will generally consider a market $\boldsymbol{P}$ paired with a deductive process $\boldsymbol{D}$. We don't much care how the deductive process works, so long as the worlds that are ruled out are in fact inconsistent, and such that every inconsistent world is ruled out eventually.

Definition 2.18 (Γ-valid deductive process)**.** *A deductive process $\boldsymbol{D}$ is **Γ-valid** if $\bigcap_n D_n$, written D_∞, is equal to the set of completions of Γ.*

As an example, let $\text{Con}_n^{\mathsf{PA}}$ be the set of all W such that there is no proof of $\bot$ with $\leq n$ characters from $\mathsf{PA} \cup \{\phi \mid W(\phi) = 1\} \cup \{\neg\phi \mid W(\phi) = 0\}$, i.e., $\text{Con}_n^{\mathsf{PA}}$ is the set of all worlds that seem consistent with PA if you only look at proofs up to length n. $\mathbf{Con}^{\mathsf{PA}}$ is a PA-valid deductive process.

2.5 Exploitation

Definition 2.19 (Worth function)**.** *The **worth function** $\text{Worth} : \mathcal{W} \to \mathbb{R}$ of a trader $\boldsymbol{t}$ trading against a market $\boldsymbol{P}$ with deductive process $\boldsymbol{D}$ is a function that takes in a pair (W, n) such that $W \in D_n$ and computes the amount of money the trader would have made up to and including day n, if all sentences were settled by W:*

$$\text{Worth}_{\boldsymbol{t}}^{\boldsymbol{P}}(W, n) := \sum_{i \leq n} \sum_{\phi \in \Lambda} t_i(\phi) \cdot (W(\phi) - P_i(\phi)) .$$

$t_i(\phi) \cdot (W(\phi) - P_i(\phi))$ is the profit of $\boldsymbol{t}$ from buying/selling shares of ϕ on day i, where $t_i(\phi)$ is the number of shares bought (if positive) or sold (if negative); $W(\phi)$ is the payout (1 or 0) according to W; and $P_i(\phi)$ is the price of ϕ-shares on day i.

Observe that a trader's plausible worth might be positive even if they have only bought and sold undecidable sentences. For example, if ϕ is undecidable and on day

n the trader bought a share of ϕ at a price of 25 cents, then there is at least one world where they have a plausible worth of 75 cents, and at least one world in which they have a plausible worth of -25 cents.

Definition 2.20 (Exploitation)**.** *A trader* $\boldsymbol{t}$ *is said to* ***exploit*** *the market* $\boldsymbol{P}$ *relative to the deductive process* $\boldsymbol{D}$ *if* $\text{Worth}_{\boldsymbol{t}}^{\boldsymbol{P}}$ *is bounded below but not above.*

In colloquial terms, a trader exploits the market if they can attain unbounded returns with only finite risk (according to the worlds that seemed plausible to the market at the time). For example, if the prices of ϕ and $\neg\phi$ are both fixed at 40 cents indefinitely, a trader can exploit the market by buying one share of each every day, because then their maximum worth will be bounded below (eventually all worlds with ϕ and $\neg\phi$ both false will be refuted) but not above (according to at least one world, it goes up by 20 cents per day).

3 The logical induction criterion

Definition 3.1 (logical induction criterion)**.** *A market* $\boldsymbol{P}$ *is said to satisfy the* ***logical induction criterion*** *with respect to a deductive process* $\boldsymbol{D}$ *if it cannot be exploited by any trader* $\boldsymbol{t}$*, i.e. if for all* $\boldsymbol{t} \in \mathcal{T}$*,*

$$\text{Worth}_{\boldsymbol{t}}^{\boldsymbol{P}} \text{ bounded below} \Rightarrow \text{Worth}_{\boldsymbol{t}}^{\boldsymbol{P}} \text{ bounded above.}$$

A market $\boldsymbol{P}$ *which meets this criterion is called a* ***logical inductor****.*

Theorem 3.1. *There exists a computable reasoning process that implements a logical inductor.*

Proof. Deferred to section 5, where we give the algorithm. □

4 Properties of Logical Inductors

Here is an intuitive argument that logical inductors perform good reasoning under logical uncertainty:

> Consider any polynomial-time method for identifying patterns in logic. If the market prices don't learn to reflect that pattern, there is a clever trader who uses that pattern to exploit the market. Thus, a logical inductor must identify all the varied patterns in logic that can be expressed as relationships between prices, and start making the market prices reflect those relationships.
>
> Furthermore, foolish traders lose their money quickly, while brilliant traders make a fortune and can have a higher trading volume. The prices that clear the market will therefore pay more attention to the brilliant traders. Thus, logical inductors learn inductively which sorts of patterns to pay more attention to, by listening more to traders that identified patterns well in the past.

We will now provide evidence for this intuitive argument, by demonstrating a number of desirable properties possessed by logical inductors. For example, we will show that logical inductors learn, in a timely manner, to assign high probability to any pattern of theorems that can be identified in polynomial time; and to use appropriate statistical summaries on sequences that appear pseudorandom; and to know their current beliefs; and to trust their future beliefs.

Our goal is to show that logical inductors recognize many different types of patterns inductively, and that they manage their uncertainty in a reasonable fashion, like any good inductive process. Of course, the uncertainty of a logical inductor stems from a lack of logical knowledge and computational resources, rather than

a lack of empirical information. Logical inductors therefore give a theory of how to inductively learning to predict the behavior of long-running programs from the observation of shorter-running ones, and so on.

We will prove some of the theorems below, but we defer a number of proofs to the extended version of this paper.

In what follows, $\boldsymbol{P}$ will always denote a computable reasoning process which implements a logical inductor relative to an associated Γ-valid deductive process $\boldsymbol{D}$.

4.1 Limit properties

The most basic properties of logical inductors is that their beliefs always converge to a coherent probability distirbution.

Theorem 4.1 (Covergence)**.** *The limit $P_\infty : \Lambda \to [0,1]$ defined by*

$$P_\infty(\phi) := \lim_{n\to\infty} P_n(\phi)$$

exists for all ϕ.

Proof. Deferred to Section 6.1. □

Theorem 4.2 (Limit Coherence)**.** *P_∞ is coherent, i.e., it gives rise to an internally consistent probability measure $\mathbb{P}$ on the set D_∞ of all worlds consistent with Γ, defined by the formula*

$$\mathbb{P}(W(\phi) = 1) := P_\infty(\phi).$$

In other words, P_∞ is a probability measure on the set of completions of Γ.

Proof. Deferred to Section 6.2. □

The limiting belief-state is non-dogmatic, in the sense that it assigns non-extreme probabilities to all undecidable sentences. In fact, the limiting probability of an undecidable sentence is bounded away from 0 and 1 by an amount proportional to its Kolmogorov complexity.

Theorem 4.3 (Occam bounds)**.** *There exists a fixed positive constant C such that for any sentence ϕ with Kolmogorov complexity $\kappa(\phi)$, if $\Gamma \nvdash \neg\phi$, then*

$$P_\infty(\phi) \geq C2^{-\kappa(\phi)},$$

and if $\Gamma \nvdash \phi$, then

$$P_\infty(\phi) \leq 1 - C2^{-\kappa(\phi)}.$$

Proof. See the extended paper. □

This means that a logical inductor can be used to do sequence prediction. A finite bitstring like 00101... can be encoded as a sentence $\neg b_1 \wedge \neg b_2 \wedge b_3 \wedge \neg b_4 \wedge b_5 \ldots$, and if the $(b_1, b_2, \ldots)$ are not mentioned in the axioms of Γ, then a logical inductor will assign this truth assignment a probability proportional to the Kolmogorov complexity of the bitstring.

Theorem 4.4 (Domination of the Universal Semimeasure)**.** *Let $(b_1, b_2, \ldots)$ be a sequence of zero-arity predicate symbols in the language of Γ not mentioned in the axioms of Γ, and let $\boldsymbol{\sigma}$ be an infinite bitstring. Define*

$$P_\infty(\boldsymbol{\sigma}_{\leq n}) = P_\infty(\text{“}(b_1 \leftrightarrow \sigma_1 = 1) \wedge (b_2 \leftrightarrow \sigma_2 = 1) \wedge \ldots \wedge (b_n \leftrightarrow \sigma_n = 1)\text{”}).$$

Let M be a universal semimeasure. Then there is some constant C such that for any finite bitstring $\boldsymbol{\sigma}_{\leq n}$,

$$P_\infty(\boldsymbol{\sigma}_{\leq n}) \geq C \cdot M(\boldsymbol{\sigma}_{\leq n}).$$

Proof. See the extended paper. □

4.2 Pattern recognition

Logical inductors learn a number of different patterns in a timely manner. For example,

Theorem 4.5 (Provability Induction)**.** *If $\boldsymbol{\phi}$ is a efficiently enumerable sequence of provable sentences, then*

$$P_n(\phi_n) \eqsim_n 1.$$

In other words, if $\boldsymbol{\phi}$ is an e.e. sequence of theorems then $\boldsymbol{P}$ learns to assign high probabilities to $\boldsymbol{\phi}$ in a timely manner.

Proof. This is an immediate corollary of Theorem 4.7. □

Consider an e.e. sequence $\boldsymbol{\phi}$ of theorems which can be generated in polynomial time, but are quite difficult to prove. Let $f(n)$ be the time at which $\boldsymbol{D}$ will prove ϕ_n, and assume that f is some fast-growing function. At any given time n, the statement ϕ_n is ever further out beyond the current deductive state D_n—it might take 1 day to prove ϕ_0, 10 days to prove ϕ_1, 100 days to prove ϕ_2, and so on. One might therefore expect that ϕ_n will also be "out of reach" for P_n, and that we have to wait until a day close to $f(n)$ before expecting the prices $P_{f(n)}(\phi_n)$ to be confident in ϕ_n. However, this is not the case!

Provability induction says that, for large n and a sequence $\boldsymbol{\phi}$ of theorems that can be efficiently enumerated, $P_n(\phi_n) > 1 - \varepsilon$, despite the fact that ϕ_n will not be confirmed deductively until a much later time $f(n)$. In other words, $\boldsymbol{P}$ inductively learns the pattern, and its prices for the $\boldsymbol{\phi}$ become accurate faster than $\boldsymbol{D}$ can computationally verify them.

> **Analogy: Ramanujan and Hardy.** Imagine that the statements $\boldsymbol{\phi}$ are being output by an algorithm that uses heuristics to generate mathematical facts without proofs, playing a role similar to the famously brilliant, often-unrigorous mathematician Srinivas Ramanujan. Then $\boldsymbol{P}$ plays the historical role of the beliefs of the rigorous G.H. Hardy who tries to verify those results according to a slow deductive process. After Hardy ($\boldsymbol{P}$) verifies enough of Ramanujan's claims ($\boldsymbol{\phi}_{\leq n}$) according to some slow deductive process ($\boldsymbol{D}$), he begins to trust Ramanujan, even if the proofs of Ramanujan's later conjectures are impossibly long, putting them ever-further beyond Hardy's current abilities to rigorously verify them. In this story, Hardy's inductive reasoning (and Ramanujan's also) outpaces his deductive reasoning.

Similar theorems hold for sequences of contradictions (if $\boldsymbol{\psi}$ is an e.e. sequence of contradictions, then $\boldsymbol{P}$ assigns probability 0 to $\boldsymbol{\psi}$ in a timely manner) and sequences of undecidable sentences that converge to a single probability, and so on.

In fact, if the logical inductor is going to learn to assign probabilities $\boldsymbol{p}$ to $\boldsymbol{\phi}$ at any point in the future, then it learns to assign those probabilities in a timely manner instead:

Theorem 4.6 (Timely adoption of bounds)**.** *Let $\boldsymbol{\phi}$ be an e.e. sequence of sentences. Then*

$$\liminf_{n\to\infty} P_n(\phi_n) = \liminf_{n\to\infty} \sup_{m\geq n} P_m(\phi_n).$$

Furthermore, the equation also holds with infimums and supremums swapped.

Proof. See the extended paper. □

For example, consider the sequence

$$\boldsymbol{\pi\text{Aeq7}} := \big(\text{“}\pi[A(\underline{n},\underline{n})] = 7\text{”}\big)_{n\in\mathbb{N}^+}$$

where $\pi[i]$ is the ith decimal digit of π and A is the Ackermann function. Each individual sentence is decidable, so the limiting probabilities are 0 for some πAeq7_n

and 1 for others. But that pattern of 1s and 0s is not efficiently enumerable (to say the least). Theorem 4.6 says that even so, if $\boldsymbol{P}$ is going to eventually learn to assign probability 10% to each πAeq7_n while it waits to learn the Ackermann numbers, then it learns to assign 10% probability to the sequence in a timely manner.

Furthermore, logical inductors do learn to assign probabilities like 10% to sequences like $\boldsymbol{\pi}$**Aeq7**. To formalize this claim, we need to formalize the idea that a sequence is "apparently random" to a reasoner. Intuitively, this notion must be defined relative to a specific reasoner and their computational limitations. After all, the digits of π are perfectly deterministic; they only appear random to a reasoner who lacks the resources to compute them.

Roughly speaking, what we will do is this. Given a sequence $\boldsymbol{\phi}$ of decidable e.e. statements, we will consider functions f that attempt to single out the true statements in the sequence. In particular, we will consider only functions f that have a runtime on the same order of complexity $\mathcal{O}(P_n)$ as the runtime of the algorithm that produces the nth price table in $\boldsymbol{P}$. We will use the shorthand $\mathcal{O}(f) = \mathcal{O}(\boldsymbol{P})$ to denote the claim that $\mathcal{O}(f(n)) = \mathcal{O}(P_n)$ for all n. We will check the limiting frequency with which they single out true ϕ_n successfully. If all of those functions converge on the same limiting accuracy, then we will say that the sequence $\boldsymbol{\phi}$ is apparently random (with respect to $\mathcal{O}(\boldsymbol{P})$), because no function can, in that runtime, do better than guessing.

Definition 4.1 ($\mathcal{O}(\boldsymbol{P})$ fuzzy subset)**.** *A* ***fuzzy subset*** $f : \mathbb{N}^+ \to [0,1]$ *is a function from natural numbers to* $[0,1]$*, such that* $\sum_n f(n) = \infty$*. If* $\mathcal{O}(f) = \mathcal{O}(\boldsymbol{P})$ *as functions of* n*, we say that* f *is* ***practical*** *relative to* $\mathcal{O}(\boldsymbol{P})$*.*

Definition 4.2 (p-pseoudorandom sequence)**.** *A efficiently enumerable sequence* $\boldsymbol{\phi}$ *of decidable sentences is called* p***-pseudorandom*** *(with respect to* $\mathcal{O}(\boldsymbol{P})$ *if, for all practical fuzzy subsets* f*,*

$$\lim_{n\to\infty} \frac{\sum_{i<n} f(i) \cdot [\Gamma \vdash \phi_i]}{\sum_{i<n} f(i)}$$

exists and is equal to p*.*

A few notes on these definitions, before proceeding. First, note that fuzzy subsets have codomain $[0,1]$. This is necessary for our proofs, because a trader can implement a fuzzy subset, but not a true subset $\mathbb{N}^+ \to \mathbb{B}$ (which is discontinuous). For purposes of intuition, it is easiest to imagine functions that always output 0 or 1, in which case each f can be interpreted as saying "maybe this is an important subset of $\boldsymbol{\phi}$ to pay attention to". The constraint that the $f(n)$ sum to ∞ ensures that each f is talking about a pattern of the whole sequence, as opposed to just deciding a finite initial sequence of $\boldsymbol{\phi}$ and picking out the true ones precisely.

Theorem 4.7 (Learning pseudorandom frequencies)**.** *Let* $\boldsymbol{\phi}$ *be an e.e. sequence of* Γ*-decidable sentences which is* p*-pseudorandom over* $\mathcal{O}(\boldsymbol{P})$*. Then*

$$P_n(\phi_n) \eqsim_n p.$$

Proof. Deferred to Section 6.3. □

This shows that logical inductors learn the right statistical summaries on sequences that are apparently pseudorandom relative to $\mathcal{O}(\boldsymbol{P})$. Note that it doesn't rule out $\boldsymbol{P}$ recognizing a pattern on sequences that are apparently pseudorandom relative to polytime; the market is allowed to be smarter than the sum of its parts.

Furthermore, logical inductors learn, in a timely manner, to make their probabilities respect linear inequalities that will hold between the truth value in the limit. For example, consider a computer program `prg` which outputs either 0, 1, or 2 on all inputs, but for which the general case cannot be proven by Γ. Theorem 4.5 says that the sequence

$$\textbf{prg012} := \left(\text{“}\underline{\texttt{prg}}(\underline{n}) = 0 \vee \underline{\texttt{prg}}(\underline{n}) = 1 \vee \underline{\texttt{prg}}(\underline{n}) = 2\text{”}\right)_{n\in\mathbb{N}^+}$$

will be learned, in the sense that $\boldsymbol{P}$ will inductively learn to assign each prg012_n a probability near 1 in a timely manner. But what about the following three individual sequences?

$$\begin{aligned}
\textbf{prg0} &:= \left(\text{“}\underline{\texttt{prg}}(\underline{n}) = 0\text{”}\right)_{n\in\mathbb{N}^+} \\
\textbf{prg1} &:= \left(\text{“}\underline{\texttt{prg}}(\underline{n}) = 1\text{”}\right)_{n\in\mathbb{N}^+} \\
\textbf{prg2} &:= \left(\text{“}\underline{\texttt{prg}}(\underline{n}) = 2\text{”}\right)_{n\in\mathbb{N}^+}
\end{aligned}$$

None of the three are purely theorems, so Theorem 4.5 does not apply. If they are utterly unpredictable in $\mathcal{O}(\boldsymbol{P})$, then Theorem 4.7 says that $\boldsymbol{P}$ will fall back on the limiting frequencies, but that tells us little in cases where there are predictable non-conclusive patterns (e.g., if `prg(i)` is more likely to output 2 when `helper(i)` outputs 17). In fact, the probabilities on the $(\text{prg0}_n, \text{prg1}_n, \text{prg2}_n)$ triplet *should* shift around over time, as $\boldsymbol{P}$ gains new knowledge about related facts and updates its beliefs. How could we tell if those intermediate beliefs were reasonable?

One way is to check their sum. If $\boldsymbol{P}$ believes that `prg(i)` $\in \{0, 1, 2\}$ and it knows how disjunction works, then it should be the case that whenever $P_n(\text{prg012}_t) \approx 1$, $P_n(\text{prg0}_t) + P_n(\text{prg1}_t) + P_n(\text{prg2}_t) \approx 1$. And this is precisely the case.

Definition 4.3 (Affine features)**.** *A* ***k-arity affine feature*** $\mathcal{AF} : [0,1]^\Lambda \to \mathbb{R}$ *is an affine function of k sentences such that*

$$\mathcal{AF}(T) = \alpha^0 + \sum_{i=1}^{k} \alpha^i T(\phi^i)$$

where the α^i *are market features and the* ϕ^i *are sentences.*

For instance, we can declare (by convention) that an affine $\mathcal{AF}$ feature is “satisfied” by a world W if $\mathcal{AF}(W) \geq 0$. Then the above example can be encoded as the two affine features

$$\begin{aligned}
&1 - T(\text{prg0}_t) - T(\text{prg1}_t) - T(\text{prg2}_t) \\
&T(\text{prg0}_t) + T(\text{prg1}_t) + T(\text{prg2}_t) - 1
\end{aligned}$$

where any world that satisfies both (by sending both to a number ≥ 0) is one in which exactly one of the sentences is true, and any price table P_n which satisfies both is one for which $P_n(\text{prg0}_t) + P_n(\text{prg1}_t) + P_n(\text{prg2}_t)$. We can show that logical inductors learn, in a timely manner, to make their probabilities satisfy any affine feature that holds in fact.

We call a sequence of affine features $\boldsymbol{\mathcal{AF}}$ “bounded” if there is some bound b such that for all n, $\sum_{i=0}^{k} |\alpha_i^k| \leq b$.

Theorem 4.8 (Affine coherence)**.**

$$\liminf_{n\to\infty} \mathcal{AF}_n(P_n) \geq \liminf_{n\to\infty} \mathcal{AF}_n(P_\infty) \geq \liminf_{n\to\infty} \inf_{W\in D_\infty} \mathcal{AF}_n(W)$$

and

$$\limsup_{n\to\infty} \mathcal{AF}_n(P_n) \leq \limsup_{n\to\infty} \mathcal{AF}_n(P_\infty) \leq \limsup_{n\to\infty} \sup_{W\in D_\infty} \mathcal{AF}_n(W)$$

Proof. See the extended paper. □

These inequalities connect truth in consistent worlds (the extrema to which consistent worlds send the affine feature) to the behavior in the limit (which is a weighted average of all consistent worlds) to the behavior on the main diagonal (which assigns probabilities that put all e.e. sequences of affine features into the right range in a timely manner).

This doesn't mean that $\boldsymbol{P}$ will *solve* difficult constraint satisfaction problems in a timely manner, it just means $\boldsymbol{P}$'s probabilities will start *respecting all linear inequalities* in a timely manner. For example, if a set of complex constraints holds between seven sequences of sentences, such that exactly three elements of each septuplet are true, but it's difficult to figure out which three. Then $\boldsymbol{P}$ will learn this pattern, and start ensuring that its probabilities on each seven-tuple sum to 3, even if it can't yet assign particularly high probabilities to the correct three.

If we watch an individual seven-tuple as $\boldsymbol{P}$ reasons, other constraints will push the probabilities on those seven sentences up and down. One sentence might be refuted and have its probability go to zero. Another might get a boost when $\boldsymbol{P}$ discovers that it's implied by a high-probability sentence. Another might take a hit when $\boldsymbol{P}$ discovers it might imply a low-probability sentence. Throughout all this, theorem 4.8 says that $\boldsymbol{P}$ will ensure that the seven probabilities always sum to ≈ 3. $\boldsymbol{P}$'s beliefs on any given day arise from this interplay of theorem recognition, statistical pattern recognition, and the satisfaction of many different constraints, inductively learned.

In the extended version of this paper, we prove stronger and more general versions of all the above results.

4.3 Self-knowledge

Logical inductors also learn to know what they know, and trust their future beliefs, in a way that avoids paradox. For starters,

Theorem 4.9 (Introspection)**.** *Let $\boldsymbol{\phi}$ be an efficiently enumerable sequence of sentences, $\boldsymbol{a}$ and $\boldsymbol{b}$ be efficiently enumerable sequences of probabilities, and $\boldsymbol{\delta}$ be an efficiently enumerable sequence of positive real numbers. Define the sequence*

$$\boldsymbol{\psi} := \left(\text{“}\underline{a}_{\underline{n}} < \underline{P}_{\underline{n}}(\underline{\phi}_{\underline{n}}) < \underline{b}_{\underline{n}}\text{”}\right)_{n \in \mathbb{N}^+}$$

of sentences that say "the probability of ϕ_n on day n is in the interval (a_n, b_n)". Note that $\boldsymbol{\psi}$ is e.e. (P_n need not be evaluated to produce ψ_n, it just needs to be written down). Then, for every $\varepsilon > 0$, the following two implications hold for all sufficiently large n:

1. *if $P_n(\phi_n) \in (a_n + \delta_n, b_n - \delta_n)$, then $P_n(\psi_n) > 1 - \varepsilon$.*
2. *if $P_n \notin (a_n - \delta_n, b_n + \delta_n)$, then $P_n(\psi_n) < \varepsilon$.*

Proof. See the extended paper. □

Roughly speaking, this says that if there is an e.e. pattern of the form "your probabilities on these sorts of sentences will be between (a, b)" then $\boldsymbol{P}$ will learn to believe that pattern iff it is true, subject to the caveat that its self-knowledge has only finite precision (i.e., if its beliefs are extremely close to a then it can't always tell which side of a they are on).

This "finite-precision self-knowledge" allows logical inductors to avoid the classic paradoxes of self-reference:

Theorem 4.10 (Liar's paradox resistance)**.** *Fix a rational $p \in (0, 1)$, and define a sequence of "liar sentences" $\boldsymbol{L^p}$ satisfying*

$$\Gamma \vdash \underline{L^p}_{\underline{n}} \leftrightarrow \left(\underline{P}_{\underline{n}}(\underline{L^p}_{\underline{n}}) < \underline{p}\right)$$

for all n. (This can be done using, e.g., Gödel's diagonal lemma.) Then

$$\lim_{n \to \infty} P_n(L^p_n) = p.$$

Proof. See the extended paper. □

A logical inductor responds to liar sentences $\boldsymbol{L^P}$ by assigning probabilities that converge on p. For example, if the liars sentences say "$\boldsymbol{P}$ will assign me a probability less than 80% on day n", then P_n (once $\boldsymbol{P}$ has learned the pattern) starts assigning probabilities extremely close to 80%—so close that polynomial-runtime traders can't tell if it's slightly above or slightly below.

To visualize this, imagine that someone who owns a high-precision brain-scanner and can read off your beliefs, asks you what probability you assign to the claim "you will assign probability <80% to this claim at precisely 10am tomorrow". As 10am approaches, what happens to your belief in this claim? If you become extremely confident that it's going to be true, then your confidence should drop. But if you become highly confident it's going to be false, then your confidence should spike. Thus, your probabilities should oscillate, pushing your belief so close to 80% that you're not quite sure which way the brain scanner will actually call it. In response to a liar's paradox, this is exactly how a logical inductor behaves, once it's learned how liar sentences work.

To go further, we need to define expected values taken with respect to $\boldsymbol{P}$, which we do as follows.

Definition 4.4 (Logically uncertain variable)**.** *A $[0,1]$-valued **logically uncertain variable** (LUV) is a sentence X with one free variable ν such that Γ proves it uniquely pins down exactly one real number in the interval $[0,1]$:*

$$\Gamma \vdash \exists! x \in [0,1] \colon X(x),$$

*where $X(x)$ denotes X with y substituted in for ν. The value $x \in [0,1]$ that makes $X(x)$ true is called the **value** of X.*

For example, $H :=$ "$\nu = 0.5$" is a LUV. As another example, $TPC :=$ "$(\nu = 1 \wedge$ Goldbach's conjecture$) \vee (\nu = 0 \wedge \neg$Goldbach's conjecture$)$" is a LUV whose value is somewhat more difficult to determine. The quantity $P(\text{"}X(0.5)\text{"})$ plays the role that $\mathbb{P}(X = 0.5)$ would play for probability distributions and traditional random variables. For the purpose of writing expressions like $P(X = x)$, $P(X > 0.2)$, and $P(2X + 3Y < Z)$, we declare that $P(X = 0.5)$ is shorthand for $P(\text{"}\forall x \colon X(x) \to x = 0.5\text{"})$, and that, in general, any relationship

$$\forall xy \ldots z \colon X(x) \wedge Y(y) \wedge \ldots \wedge Z(z) \to R(x, y, \ldots, z)$$

can be abbreviated $R(X, Y, \ldots, Z)$. For example,

- $P\left(X > 0.2\right)$ stands for $P\left(\text{"}\forall x \colon X(x) \to x > 0.2\right)\text{"}$.
- $P(2X + 3Y < Z)$ stands for $P(\text{"}\forall xyz \colon X(x) \wedge Y(y) \wedge Z(z) \to 2x + 3y < z\text{"})$.

We can now define expectations of LUVs with respect to a given price table. The first impulse is to use a Riemann sum; this is unsatisfactory, because if $\boldsymbol{P}$ has not yet figured out that a LUV X has a unique value, then it might assign high probability to X being in multiple different places in the $[0,1]$ interval, in which case the Riemann sum would not be in $[0,1]$. So instead, we define expectations using an analog of a cumulative density function:

Definition 4.5 (Expectations)**.** *For a given logical inductor $\boldsymbol{P}$, we define the **approximate expectation operator** $\mathbb{E}_k^{P_n}$ for P_n with precision k by*

$$\mathbb{E}_k^{P_n}(X) := \sum_{i=0}^{k-1} \frac{1}{k} P_n\left(X > \frac{i}{k}\right).$$

This has the desirable property that $\mathbb{E}_k^{P_n}(X) \in [0,1]$. Observe that $\mathbb{E}_k^{(\cdot)}$ is an affine feature with coefficients $1/k$ and sentences "$X > \underline{i/k}$".

We will often want to take a limit of $\mathbb{E}_k^{P_n}(X)$ as both k and n approach ∞. We hereby make the fairly arbitrary choice to focus on the case $k = n$ for simplicity, adopting the shorthand

$$\mathbb{E}_n := \mathbb{E}_n^{P_n}$$

Observe that when n is finite, $\mathbb{E}_n$ can be interpreted as a market n-feature. Observe that each LUV X has a unique value in every consistent world, which means that each X is a random variable (in the traditional sense) in the limit distribution P_∞, and that $\mathbb{E}_\infty$ is the usual expectation operator over P_∞, where $\mathbb{E}_\infty(X)$ is a mixture of the value of X in each consistent world (weighted according to P_∞).

It is easy to show that expectations work as expected, for example:

Theorem 4.11 (Linearity of expectation)**.** *Let $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ be efficiently enumerable bounded sequences of n-market features. Let $\boldsymbol{X}, \boldsymbol{Y}$, and $\boldsymbol{Z}$ be efficiently enumerable sequences of LUVs. If for all n, $Z_n = \alpha_n X_n + \beta_n Y_n$, then*

$$\alpha_n \mathbb{E}_n(X_n) + \beta_n \mathbb{E}_n(Y_n) \eqsim_n \mathbb{E}_n(Z_n).$$

Proof. See the extended paper. □

Many other properties of the expectation operator are discussed in the extended version.

We can now show that logical inductors trust their future beliefs.

Theorem 4.12 (Conservation of Expected Updates)**.** *Let $f : \mathbb{N}^+ \to \mathbb{N}^+$ be a function with runtime polynomial in $f(n)$, such that $f(n) > n$. Let $\boldsymbol{\phi}$ denote an e.e. sequence of sentences.*

$$P_n(\phi_n) \eqsim_n \mathbb{E}_n(P_{f(n)}(\phi_n)).$$

where $\mathbb{E}_n(P_{f(n)}(\phi_n))$ is shorthand for $\mathbb{E}_n(X_n)$ and X_n is the LUV "$\nu = \underline{P}_{\underline{f(n)}}(\underline{\phi}_{\underline{n}})$".

Proof. See the extended paper. □

Roughly speaking, this says that if $\boldsymbol{P}$ on day n believes that on day $f(n)$ it will believe ϕ_n with high probability, then it already believes ϕ_n with high probability today. In other words, logical inductors learn to adopt their predicted future beliefs as their current beliefs in a timely manner—they don't say "tomorrow I expect to believe that ϕ is true, but today I think it's false".

We will also show that, roughly speaking, logical inductors trust that if their beliefs change, then they must have changed for good reasons. To do this, we first define:

$$[\phi] := \text{“}(\nu = 1 \wedge \underline{\phi}) \vee (\nu = 0 \wedge \underline{\phi})\text{”},$$

so that $[\phi]$ is the LUV with the value 1 if ϕ is true and false otherwise; and

$$[x \geq p]_\delta := \text{“}\nu = \begin{cases} 0 & \text{if } \underline{x} < \underline{p} \\ \left(\underline{x} - \underline{p}\right)/\underline{\delta} & \text{if } \underline{p} \leq \underline{x} \leq \underline{p} + \underline{\delta} \\ 1 & \text{if } \underline{p} + \underline{\delta} < \underline{x}. \end{cases}\text{”}$$

so that $[x \geq p]_\delta$ is a LUV with the value 0 if $x < p$, 1 if $x > p + \delta$, and intermediate in between. Then

Theorem 4.13 (Self trust)**.** *Let $\boldsymbol{\phi}$ be an e.e. sequence of sentences. Let $\boldsymbol{\delta}$ be an e.e. sequence of positive rational numbers converging to zero. Let $\boldsymbol{p}$ be an e.e. sequence of rational probabilities. Then*

$$\mathbb{E}_n\left([\phi_n] \cdot [P_{f(n)}(\phi_n) \geq p_n]_{\delta_n}\right) \gtrsim_n p_n \cdot \mathbb{E}_n\left([P_{f(n)}(\phi_n) \geq p_n]_{\delta_n}\right).$$

Proof. See the extended paper. □

Roughly speaking, this says that if you ask $\boldsymbol{P}$ on day n for its belief in ϕ, conditional on it believing on day $f(n)$ that ϕ has probability p, it will answer with a probability at least p, regardless of whether or not its unconditional probability on p on day n is lower than p. In colloquial terms, conditional on its future beliefs changing, it expects them to have changed for good reasons.

Notice that $[P_{f(n)}(\phi_n) \geq p_n]_{\delta_n}$ is a continuous indicator on $\boldsymbol{P}$'s future beliefs, which can be interpreted as saying that theorem 4.13 only holds when $\boldsymbol{P}$ has finite-precision access to its future beliefs. Indeed, theorem 4.13 breaks down if $\boldsymbol{P}$ gets infinite-precision access to its future beliefs, and this is quite desirable. For example, let each ϕ_n be the liar sentence "$\underline{P}_{\underline{f(n)}}(\underline{\phi_n}) < 0.5$" which says that the future $P_{f(n)}$ will assign probability less than 0.5 to the sentence. Then, conditional on $P_{f(n)}(\phi_n) \geq 0.5$, P_n should believe that the probability of ϕ_n is 0. And indeed, this is what a logical inductor will do:

$$P_n(\text{“}\underline{\phi_n} \wedge (\underline{P}_{\underline{f(n)}}(\underline{\phi_n}) \geq 0.5)\text{”}) \eqsim_n 0,$$

by a trivial application of theorem 4.7 (the limiting frequency of this sequence is 0, because each sentence is disprovable). This is why theorem 4.13 uses expectations and fuzzy indicator functions: with discrete conjunctions, the result would be undesirable (and false).

Roughly speaking, theorem 4.13 says is that $\boldsymbol{P}$ attains self trust of the "if in the future I will believe x is very likely, then it must be because x is very likely" variety, if the conditional gives finite-precision access to its future beliefs. Simultaneously, $\boldsymbol{P}$ retains the ability to think it can outperform its future self's beliefs given infinite-precision access to them. If you ask "what's your probability on the liar sentence ϕ_n given that your future self believes it with probability *exactly* 0.5?" then $\boldsymbol{P}$ will answer "very low", but if you ask "what's your probability on the liar sentence ϕ_n given that your future self believes it with probability *extremely close to* 0.5?" then $\boldsymbol{P}$ will answer "roughly 0.5."

In the extended version of this paper, we prove generalized versions of the above results, and discuss a number of other subjects (such as calibration and conditional probabilities) in some depth.

5 Construction

We now give a computable algorithm for constructing a logical inductor.

5.1 Proof sketch

Imagine a reasoner writing out price tables at a rate of one per day, with the knowledge that those prices are going to be used to run a market full of polynomial time traders, and that they will be obligated to buy and sell arbitrarily many shares at the listed prices. If they can produce a sequence of price tables that satisfy the logical induction criterion, then the resulting prices, interpreted as beliefs about logic, will have the desirable properties listed above.

Our algorithm for doing this is easier to visualize if we begin with a finite case. First, notice that if there is a trader that exploits a market, then there is another trader that exploits the market while having their plausible worth bounded below by -1 (simply scale down the trades of the first trader). Thus, in the finite setting, the reasoner can take all traders, give them each \$1, and act as follows each day. First, run all traders to get their trade strategies. Second, cut their trades off if they go over-budget. Third, combine all the strategies into a single net trading strategy, which is a continuous function from the day's prices to a net trade, where the net trade lists the net volume on all (finitely many) sentences that some trader might buy/sell shares on that day.

Now what the reasoner can do is search for a price table that causes the net trade to be zero everywhere (or possibly causes net buy orders for shares priced at 1, or net sell orders for shares priced at 0). We will show that this can be done using Brouwer's fixed point theorem. If the net trade is zero everywhere, then for every share of ϕ bought by one trader, there's a share of ϕ sold by other traders, and the money in the system always gets shuffled around between traders, so the reasoner never puts any of their own money in. If the net trade purchases shares at price 1 or sells at price 0, no money flows into the system. The prices that make the net

trades literally zero can be difficult to find, so we will have the reasoner approximate those prices to within ε; the reasoner will cut ε in half each day, such that they only ever have to put in at most \$1 into their own market. Because the market starts with a finite amount of money (one dollar per trader for finitely many traders), and only a finite amount of money is ever added, the plausible worth of every individual trader must remain finite.

The situation gets a bit more complicated in the infinite case; we will do the following. The reasoner will take one additional trader under consideration each day. They will spread out a single dollar among all the traders, in a way that ensures that the nth trader has at most 2^{-n} wealth by the time they enter consideration. (This will entail giving the nth trader less than 2^{-n} to start, and setting aside a portion of the first dollar to cover trades that the nth trader makes before it is taken under consideration). Then, each day, they will find a price table that causes the net trade volume among all considered traders to be twice as close to zero as it was the day before. Thus, the market starts with a finite amount of money in it, and the market will only ever have a finite amount of money added to it, so the plausible wealth of every individual trader will be bounded above.

5.2 Proxy Traders

Fix an enumeration $(\boldsymbol{t}^1, \boldsymbol{t}^2, \ldots)$ of all traders. We begin by showing how to take the ith trader and construct a *proxy trader* that (1) has their trades cut off if a trade would cause their plausible worth to dip below -2^{-i} and (2) has their trading strategy scaled down so much that, on day i, their plausible worth is at most 2^{-i}.

For each $\boldsymbol{t}^i$, we define the sequence of market features $\boldsymbol{\alpha}^i$ recursively as follows. Each α^i_n takes a market history $\boldsymbol{P}_{\leq n}$ as input and computes the maximum value in $[0, 1]$ such that, for all $W \in D_n$,

$$\sum_{j \leq n} \sum_{\phi \in \Lambda} \left(\alpha^i_j(\boldsymbol{P}_{\leq j}) \cdot t^i_j(\phi)(\boldsymbol{P}_{\leq j})\right) (W(\phi) - P_j(\phi)) \geq -1.$$

Intuitively, α^i_n calculates the amount to scale t^i_n down to ensure that $\boldsymbol{t}^i$'s minimum worth on day n doesn't dip below -1, according to any world W which is still plausible to $\boldsymbol{D}$ on day n, and assuming that $\boldsymbol{t}^i_{<n}$ were scaled by $\boldsymbol{\alpha}^i_{<n}$.

Observe that each α^i_n is computable, because each t^i_j is non-zero on only finitely many sentences, so only finitely many combinations of truth values to those sentences need to be checked. (Recall that $\text{Plausible}^{\boldsymbol{D}}_n$ can be used to check whether a certain truth assignment remains plausible, and that it is computable.) However, the sequence $\boldsymbol{\alpha}^i$ cannot necessarily be generated in polynomial time.

Next, for each $\boldsymbol{t}^i$, define

$$\beta^i(\boldsymbol{P}_{<i}) := \frac{2^{-i}}{\max\left(1, \sum_{j<i} \alpha^i_j(\boldsymbol{P}_{\leq j}) \cdot |\boldsymbol{t}^i_j(\boldsymbol{P}_{\leq j})|\right)}$$

Multiplying all of $\boldsymbol{t}^i$'s trades by β^i has two effects. First, it scales all trades down by a factor of at least 2^i, ensuring that the plausible worth of the ith trader never dips below -2^{-i}. Second, notice that the denominator is the total trade volume of the ith trader before day i (or 1, whichever is greater). Thus, multiplying every trade by β^i ensures that the worth of the trader on the ith day is at most 2^{-i}, according to any world $W \in D_i$.

Now, given a sequence of price tables $\boldsymbol{P} = (P_1, P_2, \ldots)$ we can define a proxy trader $\boldsymbol{x}^i$ for each trader $\boldsymbol{t}^i$:

$$\boldsymbol{x}^i_n(\phi)(\boldsymbol{P}_{\leq n}) := \beta^i(\boldsymbol{P}_{<i}) \cdot \alpha^i_n(\boldsymbol{P}_{\leq n}) \cdot \boldsymbol{t}^i_n(\phi)(\boldsymbol{P}_{\leq n}).$$

Observe that $\boldsymbol{x}^i$ has the type of a trader, but is not a trader itself. Firstly, because $P_{<i}$ must be known before x^i_n can be computed, even if $n \ll i$; and secondly, because $\boldsymbol{\alpha}^i$ is not efficiently enumerable.

Note that

$$\inf_{(W,n)\in\mathcal{W}} \text{Worth}^{\boldsymbol{P}}_{\boldsymbol{x}^i}(W,n) \geq -2^{-i}, \tag{1}$$

and

$$\sup_{W\in D_i} \text{Worth}^{\boldsymbol{P}}_{\boldsymbol{x}^i}(W,i) \leq 2^{-i}, \tag{2}$$

which say that $\boldsymbol{x}^i$ never has its plausible worth dip below 2^{-i} (according to any then-plausible worlds), and that the maximum plausible worth of $\boldsymbol{x}^i$ on day i is 2^{-i}.

Observe that if there exists a trader $\boldsymbol{t}^i$ that exploits $\boldsymbol{P}$, then there also exists a proxy trader $\boldsymbol{x}^j$ (not necessarily $\boldsymbol{x}^i$) that exploits $\boldsymbol{P}$, by executing equivalent trades scaled down so far that equations (1) and (2) hold with $\alpha^i_n = 1$ for all n. Thus, it suffices to define a market $\boldsymbol{P}$ that cannot be exploited by any proxy trader $\boldsymbol{x}^k$. Because equation (1) says that the plausible worth of every proxy trader is bounded below in any market, it suffices to construct a market $\boldsymbol{P}$ such that the plausible total worth of all proxy traders $\boldsymbol{x}^k$ together is bounded above.

5.3 The Algorithm

We will use the following lemma to define $\boldsymbol{P}$:

Lemma 1 (Existence of clearing prices)**.** *For any $\varepsilon > 0$, any finite $\Lambda' \subseteq \Lambda$, and any continuous function* $\text{NetTrades} : [0,1]^{\Lambda'} \to \mathbb{R}^{\Lambda'}$*, there exists a price table P with support Λ', such that for all $W' \in \Lambda' \to \mathbb{B}$,*

$$\sum_{\phi\in\Lambda'} \text{NetTrades}^P(\phi) \cdot (W'(\phi) - P(\phi)) < \varepsilon.$$

Define P_n recursively from $\boldsymbol{P}_{<n}$ to be a finite-support price table such that for all $W \in D_n$,

$$\sum_{\phi\in\Lambda} \sum_{i\leq n} \boldsymbol{x}^i_n(\phi)(P_{\leq n}) \cdot (W(\phi) - P_n(\phi)) < 2^{-n}. \tag{3}$$

Lemma 1 says that such a point exists, by taking $\text{NetTrades}^P(\phi)$ to be the net trade volume $\sum_{i\leq n} \boldsymbol{x}^i_n(\phi)(\boldsymbol{P}_{<n}, P)$, and $\varepsilon = 2^{-n}$, and Λ' to be the (finite) set of sentences for which some $\boldsymbol{x}^i_n(\phi) \neq 0$ for $i \leq n$.

P_n can be computed by enumerating all (rational) price tables that are zero everywhere except for Λ' until finding one that satisfies (3), because the left-hand side of the inequality is computable.[4]

5.4 Proof of the Logical Inductor Criterion

We recall Theorem 3.1:

Theorem 3.1. *There exists a computable reasoning process that implements a logical inductor.*

Proof. Consider $\boldsymbol{P}$ as defined by equation (3). Observe that it is computable, and that it is a reasoning process (because each P_n has finite support). It remains to show that it is a logical inductor. By the definition of proxy traders, it suffices to show that $\text{Worth}^{\boldsymbol{P}}_{\boldsymbol{x}^k}$ is bounded above for all k. We will show that the plausible total worth of all proxy traders $\boldsymbol{x}^k$ together is bounded above by 3.

Notice that, on any day j, according to all worlds $W \in D_j$,

$$\sum_{i>j} \sum_{\phi\in\Lambda} \boldsymbol{x}^i_j(\phi) \cdot (W(\phi) - P_j(\phi)) \leq \sum_{i>j} 2^{-i} = 2^{-j}, \tag{4}$$

4. Clearly, this algorithm is designed for ease of proof, not for practicality.

by equation (2) and the fact that $\boldsymbol{D}$ is decreasing. This says all $W \in D_j$ agree that the reasoner will need to pay at most 2^{-j} in total to $\{\boldsymbol{x}^i \mid i > j\}$ for trades executed on the jth day. Additionally,

$$\sum_{i \leq j} \sum_{\phi \in \Lambda} \boldsymbol{x}^i_j(\phi) \cdot (W(\phi) - P_j(\phi)) < 2^{-j},$$

by equation (3), which says that all $W \in D_j$ agree that the reasoner will need to pay at most 2^{-j} to $\{\boldsymbol{x}^i \mid i \leq j\}$ for trades executed on the jth day. Combining these, we get

$$\sum_{i \in \mathbb{N}^+} \sum_{\phi \in \Lambda} \boldsymbol{x}^i_j(\phi) \cdot (W(\phi) - P_j(\phi)) < 2^{1-j}.$$

On each day n, summing over days $j \leq n$, we get that for all $W \in D_n$,

$$\sum_{j \leq n} \sum_{i \in \mathbb{N}^+} \sum_{\phi \in \Lambda} \boldsymbol{x}^i_j(\phi) \cdot (W(\phi) - P_j(\phi)) < \sum_{j \leq n} 2^{1-j} < 2, \tag{5}$$

i.e., it is always the case that every plausible world agrees that the reasoner won't ever need to pay out more than \$2 to the proxy traders in aggregate.

For any k and $(W, n) \in \mathcal{W}$,

$$\sum_{i \neq k} \sum_{j \leq n} \sum_{\phi \in \Lambda} \boldsymbol{x}^i_j(\phi) \cdot (W(\phi) - P_j(\phi)) \geq \sum_{i \neq k} -2^{-i} > -1, \tag{6}$$

by equation (1). This says that the set of all proxy traders except $\boldsymbol{x}^k$ have lost no more than \$1 in total. Combining (5) and (6), we see that for all k and $(W, n) \in \mathcal{W}$,

$$\sum_{j \leq n} \sum_{\phi \in \Lambda} \boldsymbol{x}^k_j(\phi) \cdot (W(\phi) - P_j(\phi)) < 3,$$

i.e., every plausible world always agrees that no proxy trader has a worth ≥ 3. Thus, $\text{Worth}^{\boldsymbol{P}}_{\boldsymbol{x}^k}$ is bounded above for all k. □

5.5 Proof of Lemma 1

We recall Lemma 1:

Lemma 1 (Existence of clearing prices)**.** *For any $\varepsilon > 0$, any finite $\Lambda' \subseteq \Lambda$, and any continuous function* $\text{NetTrades} : [0,1]^{\Lambda'} \to \mathbb{R}^{\Lambda'}$*, there exists a price table P with support Λ', such that for all $W' \in \Lambda' \to \mathbb{B}$,*

$$\sum_{\phi \in \Lambda'} \text{NetTrades}^P(\phi) \cdot (W'(\phi) - P(\phi)) < \varepsilon.$$

Proof. Define $\text{Fix} : [0,1]^{\Lambda'} \to [0,1]^{\Lambda'}$ as follows:

$$\text{Fix}(Q')(\phi) := \min\{1, \max\{0, Q'(\phi) + \text{NetTrades}^{Q'}(\phi)\}\}.$$

Fix can be interpreted as a function from the compact, convex space $[0,1]^{|\Lambda'|}$ to itself. Thus it has a fixed point F', by Brouwer's fixed point theorem.

For all $\phi \in \Lambda'$, since F' is a fixed point, one of the following is true:

- $\text{NetTrades}^{F'}(\phi) = 0$
- $\text{NetTrades}^{F'}(\phi) > 0$ and $F'(\phi) = 1$
- $\text{NetTrades}^{F'}(\phi) < 0$ and $F'(\phi) = 0$

It follows that $\text{NetTrades}^{F'}(\phi) \cdot (W'(\phi) - F'(\phi)) \leq 0$ for all $\phi \in \Lambda'$ and $W' : \Lambda' \to \mathbb{B}$. F' is almost the price table we need, except that it is not necessarily rational. However,

$$\sum_{\phi \in \Lambda'} \text{NetTrades}^{Q'}(\phi) \cdot (W'(\phi) - Q'(\phi))$$

is continuous in Q', so for any $\varepsilon > 0$ there is some rational $P' : \Lambda' \to [0,1]$ close enough to F' such that for all $W' : \Lambda' \to \mathbb{B}$,

$$\sum_{\phi \in \Lambda'} \text{NetTrades}^{P'}(\phi) \cdot (W'(\phi) - P'(\phi)) < \varepsilon.$$

Define $P(\phi)$ to be $P'(\phi)$ if $\phi \in \Lambda'$ and 0 otherwise. □

6 Selected proofs

In this section, we will prove theorems 4.1, 4.2, and 4.7. The remaining proofs (and proofs of many other properties) can be found in the extended version of the paper.

It will be useful for describing continuous trading strategies to have a a "soft" indicator function for events. We define a function $\text{Ind}_\delta(\,\cdot > p)$ of a real number x by

$$\text{Ind}_\delta(x > p) := \begin{cases} 0 & \text{if } x \leq p \\ \dfrac{x-p}{\delta} & \text{if } \quad p < x \leq p + \delta \\ 1 & \text{if } \quad\quad p + \delta < x. \end{cases}$$

This is the soft indicator function for the event $x > p$, which has value 0 off the event, is linear within δ of the event, and 1 otherwise. Likewise, we define the continuous indicator of $x < p$ to be:

$$\text{Ind}_\delta(x < p) = \begin{cases} 1 & \text{if } x \leq p - \delta \\ \dfrac{p-x}{\delta} & \text{if } \quad p - \delta < x \leq p \\ 0 & \text{if } \quad\quad p < x. \end{cases}$$

NB: the meaning of δ is different depending on the direction of the inequality: $\text{Ind}_\delta(x > p)$ is linear (and increasing) on the interval $[p, p + \delta]$, while $\text{Ind}_\delta(x < p)$ is linear (and decreasing) on the interval $[p - \delta, p]$.

6.1 Convergence

Recall Theorem 4.1:

Theorem 4.1 (Covergence)**.** *The limit $P_\infty : \Lambda \to [0,1]$ defined by*

$$P_\infty(\phi) := \lim_{n \to \infty} P_n(\phi)$$

exists for all ϕ.

Proof of Theorem 4.1. Suppose by way of contradiction that the limit P_∞ does not exist. Then, for some sentence ϕ and some rational numbers $p \in [0,1]$ and $\varepsilon > 0$, we have that $P_n(\phi) < p - \varepsilon$ infinitely often and $P_n(\phi) > p + \varepsilon$ infinitely often. We will show that $\boldsymbol{P}$ can be exploited, contrary to the logical induction criterion.

Roughly speaking, the trader will work as follows. Wait for a time step n on which the market assigns a low price $P_n(\phi) < p - \varepsilon$ to the sentence ϕ. Since this price is guaranteed to rise above $p + \varepsilon$, simply buy a share in ϕ at time n, and then sell back that ϕ-share when the price is high. Since the trader will then hold no net shares in ϕ, their worth is not affected by whether or not ϕ is true. On the other hand, they will have made a profit by buying low and selling high: they spent at

most $p-\varepsilon$ buying a ϕ-share, and then made at least $p+\varepsilon$ selling a ϕ-share. Since the prices $P_n(\phi)$ fluctuate forever, they will continue making money forever and thus exploit the market.

We will define a trader $\boldsymbol{t}$ that executes a strategy similar to this one, and hence exploits the market $\boldsymbol{P}$ if $\lim_{n\to\infty} P_n(\phi)$ does not converge. To do this, there are two technicalities we must deal with. First is that the strategy outlined above is a discontinuous function of the market prices $P_n(\phi)$, and therefore is not permitted. This is relatively easy to fix using soft indicator functions described above.

The second technicality is more subtle. Suppose we define our trader to buy ϕ-shares whenever their price $P_n(\phi)$ is low, and sell them back whenever their price is high. Then it is possible that the trader makes the following trades in sequence against the market $\boldsymbol{P}$: buy 10 ϕ-shares on consecutive days, then sell 10 ϕ-shares; then buy 100 ϕ-shares consecutively, and then sell them off; then buy 1000 ϕ-shares, then sell them off; and so on. Although this trader makes profit on each batch, it always spends more on the next batch, taking larger and larger risks (relative to the remaining plausible worlds). In short, this trader is not tracking its budget, and so may have unboundedly negative Worth. We will fix this problem by having our trader $\boldsymbol{t}$ track how many net ϕ-shares it has bought, and not buying too many, thereby maintaining bounded risk. This will be sufficient to prove the theorem.

Definition of the trader $\boldsymbol{t}$ Let the quantity

$$\text{Shares}_n^\phi := \sum_{i<n} t_i(\phi)$$

denote the total net number of ϕ-shares that $\boldsymbol{t}$ has purchased before the current time n. Now we define the trader $\boldsymbol{t}$ to output the trading strategy t_n on time n defined by

$$\begin{aligned} t_n(\phi) := (1-\text{Shares}_n^\phi) \times \text{Ind}_{\varepsilon/2}(P_n(\phi) < p-\varepsilon/2) \\ - \text{Shares}_n^\phi \times \text{Ind}_{\varepsilon/2}(P_n(\phi) > p+\varepsilon/2), \end{aligned}$$

and $t_n(\gamma) := 0$ for other sentences $\gamma \neq \phi$. Note that t_n is computable with runtime polynomial in n. Furthermore, the trading strategies t_n assign a single market n-feature to ϕ, as required; thus $\boldsymbol{t}$ is a well-defined trader.[5]

In words, $\boldsymbol{t}$ buys some ϕ-shares whenever $P_n(\phi) < p-\varepsilon/2$, up to 1 share when $P_n(\phi) < p-\varepsilon$; and sells some ϕ-shares whenever $P_n(\phi) > p+\varepsilon/2$, up to 1 share when $P_n(\phi) > p+\varepsilon$. These trades are scaled down according to the number Shares_n^ϕ of ϕ-shares $\boldsymbol{t}$ has bought in total. If $\boldsymbol{t}$ has bought a full ϕ-share to date that has not been balanced out by selling a ϕ-share, then $1-\text{Shares}_n^\phi = 0$, so $\boldsymbol{t}$ will not buy any more ϕ-shares; and $\text{Shares}_n^\phi = 1$, so if given the opportunity $\boldsymbol{t}$ would sell up to a full share in ϕ. Likewise, if $\boldsymbol{t}$ has bought 0 net shares in ϕ to date, then it will buy up to 1 full ϕ-share if given the opportunity by a low market price $P_n(\phi)$, but will not sell any shares in ϕ. Thus we have shown that for any time step n, our trader $\boldsymbol{t}$ has purchased a net total of $\text{Shares}_n^\phi \in [0,1]$ shares in ϕ.

Analyzing $\text{Worth}_{\boldsymbol{t}}^{\boldsymbol{P}}$. We will now lower bound the Worth of $\boldsymbol{t}$ against the market $\boldsymbol{P}$ over the deductive process $\boldsymbol{D}$.

In words: we focus just on ϕ-shares, as $\boldsymbol{t}$ doesn't trade on other sentences. As time goes on, since $\text{Shares}_n^\phi \in [0,1]$, $\boldsymbol{t}$ holds at most 1 net share in ϕ. Thus the payouts and the costs from the surplus ϕ-shares held by $\boldsymbol{t}$ contribute little to the total profit. If $\boldsymbol{t}$ has bought $k/2$ shares and sold $k/2$ shares in ϕ, then by the definition of when $\boldsymbol{t}$ buys and sells, $\boldsymbol{t}$ has made a profit of a least $(k/2)(p+\varepsilon/2-(p-\varepsilon/2)) = k\varepsilon/2$. The payouts from those ϕ-shares cancel each other out.

5. Note that $\boldsymbol{t}$ is defined making reference to the values of its previous trades $t_i(\phi)$. This can be done efficiently in general, by computing $\boldsymbol{t}$'s past trades inside $\boldsymbol{t}$'s trading strategy using dynamic programming. For details, see the extended paper.

Formally, for any time n and any world $W \in D_n$,

$$\begin{aligned}\text{Worth}_{\boldsymbol{t}}^{\boldsymbol{P}}(W,n) &:= \sum_{i\le n}\sum_{\psi\in\Lambda} t_i(\psi)\cdot(W(\psi)-P_i(\psi))\\ &= \sum_{i\le n} t_i(\phi)\cdot(W(\phi)-P_i(\phi))\end{aligned}$$

since $\boldsymbol{t}$ only makes non-zero trades on ϕ;

$$= \left(\text{Shares}_{n+1}^{\phi}\cdot W(\phi)\right) - \sum_{i\le n} t_i(\phi)\cdot P_i(\phi)$$

since $\boldsymbol{t}$ holds Shares_n^{ϕ}-many net ϕ-shares;

$$\begin{aligned}&\ge \text{Shares}_{n+1}^{\phi}\cdot 0 + \sum_{\substack{i\le n\\ P_i(\phi)<p-\varepsilon/2}} t_i(\phi)\cdot(-P_i(\phi))\\ &\qquad + \sum_{\substack{i\le n\\ P_i(\phi)>1+\varepsilon/2}} (-t_i(\phi))\cdot P_i(\phi)\\ &\ge \sum_{\substack{i\le n\\ P_i(\phi)<p-\varepsilon/2}} t_i(\phi)\cdot(-(p-\varepsilon/2))\\ &\quad + \sum_{\substack{i\le n\\ P_i(\phi)>p+\varepsilon/2}} (-t_i(\phi))\cdot(p+\varepsilon/2)\end{aligned}$$

since $t_i(\phi) > 0$ iff $P_i(\phi) < 1-\varepsilon/2$, and $t_i(\phi) < 0$ iff $P_i(\phi) > 1+\varepsilon/2$;

$$\begin{aligned}&= \sum_{i\le n} |t_i(\phi)|\cdot\varepsilon/2 - \sum_{i\le n} t_i(\phi)\cdot p\\ &= \sum_{i\le n} |t_i(\phi)|\cdot\varepsilon/2 - \text{Shares}_{n+1}^{\phi}\cdot p\\ &\ge -p + \sum_{i\le n}|t_i(\phi)|\cdot\varepsilon/2.\end{aligned}$$

Thus, the worth of $\boldsymbol{t}$ at time n is roughly $\varepsilon/2$ times the total trading volume $|\boldsymbol{t}_{\le n}|$ up until time n. In particular, $\text{Worth}_{\boldsymbol{t}}^{\boldsymbol{P}}$ is bounded below by $-p$. Futhermore, by supposition, $P_i(\phi) < 1-\varepsilon$ and then $P_j(\phi) > 1+\varepsilon$ for infinitely many i and infinitely many $j > i$. But that means that infinitely often, our trader $\boldsymbol{t}$ will have purchased a full ϕ-share (i.e. $\text{Shares}_n^{\phi} = 1$), and then sold back a full ϕ-share (i.e. $\text{Shares}_n^{\phi} = 0$), and so on, making $\varepsilon/2$ profit each time.

That is, the sum $\sum_{i\le n}|t_i(\phi)|$ diverges to ∞ as $n\to\infty$. Thus $\boldsymbol{t}$ has worth against $\boldsymbol{P}$ lower bounded but not upper bounded, and therefore exploits the market $\boldsymbol{P}$. This contradicts that $\boldsymbol{P}$ is a logical inductor; therefore, in fact the limit $P_\infty(\phi)$ must exist. □

6.2 Limit Coherence

Recall Theorem 4.2:

Theorem 4.2 (Limit Coherence)**.** *P_∞ is coherent, i.e., it gives rise to an internally consistent probability measure $\mathbb{P}$ on the set D_∞ of all worlds consistent with Γ, defined by the formula*

$$\mathbb{P}(W(\phi)=1) := P_\infty(\phi).$$

In other words, P_∞ is a probability measure on the set of completions of Γ.

Proof of Theorem 4.2. By Convergence (Theorem 4.1), the limit $P_\infty(\phi)$ exists for all sentences $\phi \in \Lambda$. Therefore, $\mathbb{P}(W(\phi) = 1) := P_\infty(\phi)$ is well-defined as a function of basic subsets of D_∞.

Gaifman (1964) shows that $\mathbb{P}$ extends to a probability measure over D_∞ so long as the following three implications hold for all sentences ϕ an ψ:

- If $\Gamma \vdash \phi$, then $P_\infty(\phi) = 1$.
- If $\Gamma \vdash \neg\phi$, then $P_\infty(\phi) = 0$.
- If $\Gamma \vdash \neg(\phi \wedge \psi)$, then $P_\infty(\phi \vee \psi) = P_\infty(\phi) + P_\infty(\psi)$.

Since the three conditions are quite similar in form, we will prove them simultaneously using three exemplar traders and parallel arguments.

Suppose for contradiction that one of these implications fails to hold by some amount ε. For example, suppose that $P_\infty(\phi \vee \psi) = P_\infty(\phi) + P_\infty(\psi) - \varepsilon$, even though Γ proves that $\neg(\phi \wedge \psi)$. Intuitively, shares in $\phi \vee \psi$ are underpriced by around ε. If we wait for P_n to approximately converge, and then buy a $(\phi \vee \psi)$-share and sell a ϕ-share and a ψ-share, we have made a profit of about ε. We will have to pay out on at most one of the ϕ-share and the ψ-share, and if we do, we will also recieve a payout from the $(\phi \vee \psi)$-share. In this way we can exploit the market $\boldsymbol{P}$ repeatedly, for unbounded gains at no risk.

Definition of the traders $\boldsymbol{t^\phi}$, $\boldsymbol{t^\psi}$, $\boldsymbol{t^{\phi\vee\psi\geq}}$, and $\boldsymbol{t^{\phi\vee\psi\leq}}$. Suppose that one of the three conditions is violated by ε, i.e.

- $\Gamma \vdash \phi$, but $P_\infty(\phi) < 1 - \varepsilon$.
- $\Gamma \vdash \neg\phi$, but $P_\infty(\phi) > \varepsilon$.
- $\Gamma \vdash \neg(\phi \wedge \psi)$, but $P_\infty(\phi \vee \psi) < P_\infty(\phi) + P_\infty(\psi) - \varepsilon$.
- $\Gamma \vdash \neg(\phi \wedge \psi)$, but $P_\infty(\phi \vee \psi) > P_\infty(\phi) + P_\infty(\psi) + \varepsilon$.

Since the limit P_∞ exists, there is some sufficiently large time s_ε such that for all $n > s_\varepsilon$, the inequality holds for P_n; e.g., for all $n > s_\varepsilon$, $P_n(\phi\vee\psi) > P_n(\phi)+P_n(\psi)+\varepsilon$.

Furthermore, since $\boldsymbol{D}$ is a Γ-valid deductive process, for some sufficiently large s_Γ and all $n > s_\Gamma$, all worlds W in the deductive state D_n satisfy the appropriate logical constraint on the sentences ϕ and ψ. For example, eventually all plausible worlds assign 0 to ϕ if $\Gamma \vdash \neg\phi$; and if $\Gamma \vdash \neg(\phi \wedge \psi)$, then eventually all worlds assign 1 to at most one of ϕ and ψ, and if either are assigned 1 then so is $\phi \vee \psi$.

Let $s := \max\{s_\varepsilon, s_\Gamma\}$. We now define traders that will exploit the market $\boldsymbol{P}$. All of the following traders will make no non-zero trades for $n \leq s$, and will make non-zero trades only for the sentences explicitly mentioned. For $n > s$, we define

$$t_n^\phi(\phi) := 1$$

to be the trader $\boldsymbol{t}^\phi$ that attempts to ensure $P_\infty(\phi) = 1$ by buying ϕ-shares;

$$t_n^{\neg\phi}(\phi) := -1$$

to be the trader $\boldsymbol{t}^{\neg\phi}$ that attempts to ensure $P_\infty(\phi) = 0$ by selling ϕ-shares;

$$\begin{aligned} t_n^{\phi\vee\psi\geq}(\phi) &:= -1 \\ t_n^{\phi\vee\psi\geq}(\psi) &:= -1 \\ t_n^{\phi\vee\psi\geq}(\phi \vee \psi) &:= 1 \end{aligned}$$

to be the trader $\boldsymbol{t}^{\phi\vee\psi\geq}$ that attempts to ensure $P_\infty(\phi \vee \psi) \geq P_\infty(\phi) + P_\infty(\psi)$; and

$$\begin{aligned} t_n^{\phi\vee\psi\leq}(\phi) &:= 1 \\ t_n^{\phi\vee\psi\leq}(\psi) &:= 1 \\ t_n^{\phi\vee\psi\leq}(\phi \vee \psi) &:= -1 \end{aligned}$$

to be the trader $\boldsymbol{t}^{\phi\vee\psi\leq}$ that attempts to ensure $P_\infty(\phi\vee\psi) \leq P_\infty(\phi)+P_\infty(\psi)$. These traders all run in constant time; the constant s can be hard-coded at constant cost.

Analyzing the worth of the traders. Consider the trader $\boldsymbol{t}^\phi$ that attempts to ensure $P_\infty(\phi) = 1$ by buying ϕ-shares. If in fact $\Gamma \vdash \phi$, then we have, for any time step n and any world $W \in D_n$ plausible at time n:

$$\begin{aligned}\text{Worth}^{\boldsymbol{P}}_{\boldsymbol{t}^\phi}(W, n) &:= \sum_{i \leq n} \sum_{\gamma \in \Lambda} t_i^\phi(\gamma) \cdot (W(\gamma) - P_i(\gamma)) \\ &= \sum_{s < i \leq n} t_i^\phi(\phi) \cdot (W(\phi) - P_i(\phi))\end{aligned}$$

since $\boldsymbol{t}^\phi$ only makes non-zero trades on ϕ and after time s;

$$\geq \sum_{s < i \leq n} 1 \cdot (1 - (1 - \varepsilon))$$

since by definition of s, $W(\phi) = 1$ and $P_i(\phi) < 1 - \varepsilon$;

$$= \varepsilon(n - s).$$

In words, $\boldsymbol{t}^\phi$ spends at most $1 - \varepsilon$ on a ϕ-share and immediately gets a pay out of 1, for a net gain of at least ε.

The analysis for $\boldsymbol{t}^{\neg\phi}$ is almost identical, so we will not belabor it; at every time $n > s$, $\boldsymbol{t}^{\neg\phi}$ sells a ϕ-share for at least ε, and if $\Gamma \vdash \neg\phi$, then in every plausible world that share is worthless and $\boldsymbol{t}^{\neg\phi}$ has made a clear profit.

Now consider the trader $\boldsymbol{t}^{\phi\vee\psi\geq}$ that attempts to ensure $P_\infty(\phi \vee \psi) \geq P_\infty(\phi) + P_\infty(\psi)$ by buying shares in $\phi \vee \psi$ and selling ϕ-shares and ψ-shares. If in fact $\Gamma \vdash \neg(\phi \wedge \psi)$, then we have, for any time step $n$and any world $W \in D_n$ plausible at time n:

$$\begin{aligned}\text{Worth}^{\boldsymbol{P}}_{t^{\phi\vee\psi\geq}}(W, n) &:= \sum_{i \leq n} \sum_{\gamma \in \Lambda} t_i^{\phi\vee\psi\geq}(\gamma) \cdot (W(\gamma) - P_i(\gamma)) \\ &= \sum_{s < i \leq n} \Big(t_i^{\phi\vee\psi\geq}(\phi) \cdot (W(\phi) - P_i(\phi)) \\ &\qquad + t_i^{\phi\vee\psi\geq}(\psi) \cdot (W(\psi) - P_i(\psi)) \\ &\qquad + t_i^{\phi\vee\psi\geq}(\phi \vee \psi) \cdot (W(\phi \vee \psi) - P_i(\phi \vee \psi)) \Big)\end{aligned}$$

since $\boldsymbol{t}^\phi$ only makes non-zero trades on ϕ, ψ, and $\phi \vee \psi$, and only after time s;

$$\begin{aligned}&= \sum_{s < i \leq n} (W(\phi \vee \psi) - W(\phi) - W(\psi) \\ &\qquad - P_i(\phi \vee \psi) + P_i(\phi) + P_i(\psi)) \\ &\geq \sum_{s < i \leq n} \varepsilon \\ &= \varepsilon(n - s),\end{aligned}$$

since $W(\phi \vee \psi) - W(\phi) - W(\psi) = 0$ and by assumption on the prices P_i after time s. The analysis for $\boldsymbol{t}^{\phi\vee\psi\leq}$ is almost identical.

Exploiting the market $\boldsymbol{P}$. The above analysis shows that for each of the traders $\boldsymbol{t}^\phi$, $\boldsymbol{t}^\psi$, $\boldsymbol{t}^{\phi\vee\psi\geq}$, and $\boldsymbol{t}^{\phi\vee\psi\leq}$, if the condition on P_∞ that they enforce is not satisfied, then at every time step n their worth against $\boldsymbol{P}$ is $\varepsilon(n-s)$. As $n \to \infty$, the worth of such a trader is unbounded above, and is bounded below by 0. Thus, if $\boldsymbol{P}$ failed to satisfy any of the coherence conditions, then the corresponding trader would exploit $\boldsymbol{P}$, contradicting that $\boldsymbol{P}$ is a logical inductor. Therefore P_∞ is coherent. □

6.3 Learning Pseudorandom Frequencies

Recall Theorem 4.7:

Theorem 4.7 (Learning pseudorandom frequencies)**.** *Let $\boldsymbol{\phi}$ be an e.e. sequence of Γ-decidable sentences which is p-pseudorandom over $\mathcal{O}(\boldsymbol{P})$. Then*

$$P_n(\phi_n) \eqsim_n p.$$

Proof of Theorem 4.7. Assume the theorem did not hold. Then, intuitively, $\boldsymbol{P}$ repeatedly underprices the ϕ_n. A trader can buy ϕ_n-shares whenever their price goes below $p - \varepsilon$. By the assumption that the truth values of the ϕ_n are pseudorandom, roughly p proportion of the shares will pay out. Since the trader only pay at most $p - \varepsilon$ per share, on average they make ε on each trade, so over time we exploit the market.

We will have to make these trades continuous. Furthermore, as in the proof of Theorem 4.1, we will need to budget our trader to prevent it from investing more and more in sentences ϕ_n that may take a very long time to pay out.

Suppose for contradiction that $\boldsymbol{\phi}$ is an e.e. sequence of Γ-decidable sentences such that for every practical fuzzy subset f,

$$\lim_{n\to\infty} \frac{\sum_{i<n} f(i) \cdot [\Gamma \vdash \phi_i]}{\sum_{i<n} f(i)} = p;$$

but nevertheless, for some $\varepsilon > 0$ and infinitely many n, we have (say)

$$P_n(\phi_n) < p - \varepsilon.$$

(The argument for the case where $P_n(\phi_n) > p + \varepsilon$ infinitely often will be the same, *mutatis mutandis*, and one of these two cases must obtain.)

Definition of the trader $\boldsymbol{t}$. Say that a sentence ϕ is *settled* at time n if every world $W \in \boldsymbol{D}_n$ assigns the same value to ϕ. Because $\boldsymbol{D}$ is Γ-valid and every ϕ_n is decidable, every ϕ_n is eventually settled. Since the deductive process $\boldsymbol{D}$ is computable, there is a function $S : \Lambda \to \mathbb{N}^+ \to \{0, 1\}$, computable in time polynomial in n, that approximates the set of sentences by assigning 0 to settled sentences and 1 to possibly unsettled sentences. That is,

- for any ϕ and any n, if $S(n, \phi) = 0$ then ϕ is settled at time n;
- for any ϕ and any $m > n$, if $S(n, \phi) = 0$ then $S(m, \phi) = 0$; and
- for any ϕ, for some sufficiently large n, $S(n, \phi) = 0$.

Note that S may assign 1 to settled sentences (this allows S to be efficiently computed, even if the deductive process $\boldsymbol{D}$ is not).

Our trader $\boldsymbol{t}$ will never have more than one share unsettled at any given time, in order to avoid going into debt by purchasing ever more unsettled investments. Formally, we define

$$\beta_n := 1 - \sum_{i<n} S(\phi_i, n) t_i(\phi_i)$$

to be the fraction of $\boldsymbol{t}$'s budget of 1 that is not tied up in possibly unsettled ϕ_i-shares. Then we define $\boldsymbol{t}$ at time n to be

$$t_n(\phi_n) := \beta_n \operatorname{Ind}_{\varepsilon/2}(P_n(\phi_n) < p - \varepsilon/2),$$

and $t_n(\gamma) := 0$ for other sentences $\gamma \neq \phi$. Note that this t_n can be computed in polynomial time in n, using S. [6] In words, t_n uses some fraction of its free budget β_n to buy ϕ_n-shares, depending on to what extent they are priced below $p - \varepsilon/2$. In particular, all the β_n and all the $t_n(\phi)$ are in $[0, 1]$.

6. As mentioned in a previous footnote, $\boldsymbol{t}$ is defined using the values of its previous trades $t_i(\phi)$; this can be implemented efficiently.

$\boldsymbol{t}$ is a practical fuzzy subset. Now we show that the sequence of trades $t_n(\phi_n)$ made by our trader $\boldsymbol{t}$ against the market $\boldsymbol{P}$ forms a practical fuzzy subset. Since $\boldsymbol{t}$ is efficiently computable and $t_n(\phi_n) \in [0,1]$ for all n, it remains to show that

$$\sum_n t_n(\phi_n) = \infty.$$

That is, we want to show that $\boldsymbol{t}$ mades trades will unbounded total volume. Intuitively, every trade that $\boldsymbol{t}$ makes is eventually settled according to S, after which point β_n is large again, and $\boldsymbol{t}$ continues buying ϕ_n-shares whenever $P_n(\phi_n) < p - \varepsilon/2$. Formally, suppose this were not the case, so that for some sufficiently large m,

$$\sum_{n>m} t_n(\phi_n) < 1/2.$$

Then by definition of S, for some sufficiently large m', $S(\phi_i, m') = 0$ for all $i \leq m$. At that point, for any $n > m'$, we have

$$\begin{aligned}\beta_n &:= 1 - \sum_{i<n} S(\phi_i, n) t_i(\phi_i) \\ &1 - \sum_{m'<i<n} S(\phi_i, n) t_i(\phi_i) \\ &\geq 1/2.\end{aligned}$$

Then, by the earlier supposition on $\boldsymbol{P}$, for some $n > m'$ we have $P_n(\phi_n) < p - \varepsilon$, at which point

$$t_n(\phi_n) := \beta_n \operatorname{Ind}_{\varepsilon/2} = (P_n(\phi_n) < p - \varepsilon/2) \geq 1/2.$$

This contradicts the supposition that $\sum_n t_n(\phi_n)$ is bounded, so in fact the sum is infinite as desired, i.e. $\boldsymbol{t}$ is a practical fuzzy subset.

Analyzing the worth of $\boldsymbol{t}$. Now, by definition of $\boldsymbol{\phi}$ being a pseudorandom sequence, we have that

$$\lim_{n\to\infty} \frac{\sum_{i<n} t_i(\phi_i) \cdot [\Gamma \vdash \phi_i]}{\sum_{i<n} t_i(\phi_i)} = p.$$

Thus, for all sufficiently large n,

$$\sum_{i<n} t_i(\phi_i) \cdot [\Gamma \vdash \phi_i] \geq (p - \varepsilon/4) \sum_{i<n} t_i(\phi_i).$$

The sum on the left is, roughly, the value of all the ϕ_i-shares held by $\boldsymbol{t}$ before time n, and the sum on the right is the total number of shares purchased by $\boldsymbol{t}$. Indeed, at time n, all but at most 1 of the shares held by $\boldsymbol{t}$ has been settled, so there is a discrepancy on the left hand side of at most 1.

By definition of $\boldsymbol{t}$, the total price paid by $\boldsymbol{t}$ for its purchases before time n is at most

$$\sum_{i<n} t_i(\phi_i)(p - \varepsilon/2).$$

Thus, the total worth of $\boldsymbol{t}$ against the market $\boldsymbol{P}$ before time n is at least

$$-1 + (p - \varepsilon/4 - (p - \varepsilon/2)) \sum_{i<n} t_i(\phi_i) \simeq_n (\varepsilon/4) \sum_{i<n} t_i(\phi_i).$$

Thus $\operatorname{Worth}_{\boldsymbol{t}}^{\boldsymbol{P}}(W, n)$ is bounded below. Further, since $\boldsymbol{t}$ makes trades of unbounded volume, i.e. $\sum_n t_n(\phi_n) = \infty$, $\operatorname{Worth}_{\boldsymbol{t}}^{\boldsymbol{P}}(W, n)$ grows unboundedly. Therefore $\boldsymbol{t}$ exploits the market $\boldsymbol{P}$. This contradicts that $\boldsymbol{P}$ is a logical inductor, so in fact we must have that

$$\lim_{n\to\infty} P_n(\phi_n) = p.$$

□

7 Discussion

We have proposed the *logical induction criterion* as a criterion for beliefs about logical facts (represented explicitly as a series of price tables), and demonstrated that logical inductors have many desirable properties when it comes to reasoning about logical facts. For example, logical inductors learn (in a timely manner) to assign probabilities near 1 to any sequence of theorems that can be produced in polynomial time, which often requires outpacing the underlying deductive process. Similarly, logical inductors learn to assign probabilities near 0 to all efficiently enumerable sequences of contradictions, and they learn to assign probabilities that respect linear inequalities that will hold in the limit (such as ensuring that the probabilities of "`prg(i)`=0" and "`prg(i)`=1" sum to no more than 1, and so on), and they learn to fall back on appropriate statistical summaries in the face of pseudorandom sequences. They also learn to know their current beliefs and trust their future beliefs, and their beliefs are fully coherent in the limit.

We have presented a computable algorithm that demonstrates that it is possible to meet the logical induction criterion. Clearly, the algorithm presented in Section 5 is not intended for practical use. On the contrary, logical induction is intended to serve as a theoretical account of how to learn inductively to predict unknown logical facts (such as the behavior of a long-running computer program) from known ones (such as the observation of shorter-running computer programs). In this way, it is analogous to Solomonoff's theory of inductive inference (Solomonoff 1964), which provides a theoretical account of how to use Bayesian probability theory to learn inductively to predict unknown empirical facts (the next observation that will be produced by an environment) from known ones (the history of observations produced by the environment).

As an analogy, in the field of artificial intelligence (AI), Bayesian probability theory has provided inspiration for a wide array of AI algorithms, from Belief Propagation (Pearl 1988) to Auto-Encoding Variational Bayes (Kingma and Welling 2013). Indeed, while perfect Bayesian reasoning is intractable in practice, the idiom of representing beliefs with probabilities and responding to evidence in an approximately Bayesian fashion is ubiquitous throughout the field of AI. Similarly, while Solomonoff's theory of inductive inference is uncomputable, the idea of consulting a set of experts, rewarding them for predicting the data well, and penalizing them for complexity, is a ubiquitous idiom for predictive analytics. Logical induction aspires to provide an analogous theoretical foundation for algorithms that are predicting logical facts (such as the behavior of computations that are too expensive to run).

The authors are particularly interested in tools that help AI scientists attain novel robustness and reliability guarantees in settings where statistical guarantees are currently hard to come by. This includes settings where AI systems have to reason about computations that are too expensive to run, as discussed by Fallenstein and Soares (2015), and also settings where AI systems have to reason about *other* agents that are logically uncertain. For example, consider an AI system attempting to learn a human's goals, and imagine it observing the human make a losing chess move. Such a system should not conclude that the human wanted to lose; it should instead have a model of the human being logically uncertain, with beliefs constrained by deductive limitations. (For further discussion on this topic, refer to Russell [2014] and Li et al. [2010].) In settings such as these, we expect logical induction to help by providing an early theoretical framework for studying deductively limited (but otherwise rational) reasoners.

The authors believe that the logical induction criterion captures a sizable portion of what it means to manage uncertainty well in the face of deductive limitations, in the same way that the "no Dutch book" criteria of Ramsey (1931) and de Finetti (1979) capture a sizable portion of what it means to manage uncertainty well in the face of empirical uncertainty. Roughly speaking, logical inductors learn to recognize all patterns that can be found in polynomial time; and any reasoning process that does not satisfy the logical inductor criterion can (by definition) be exploited by some stock trader that runs in polynomial time.

Nevertheless, logical induction leaves a number of questions wide open. For

example, logical induction does not obviously give good answers to questions about counterpossibles, such as "what would math be like if Fermat's last theorem were false?" While these questions may seem ridiculous at first glance, they are important when it comes to developing a decision theory that is suitable for deductively limited reasoners, where one may want to ask "what will happen in the world if `agent(n)=action3`?" and have some confidence that the answer will be reasonable even if `agent(n)` does not in fact equal `action3`.

Finally, there is the question of whether logical inductors behave well in practice. One key insight of the algorithm in Section 5 is that it is possible for a limited reasoner to construct reasonable beliefs about logical facts by consulting a collection of "stock traders", where each trader watches for (and attempts to profit from) a certain type of price inefficiency when the beliefs are interpreted as a price table. "Reasonable" beliefs (relative to a collection of traders) are then any prices that make the market clear. The authors expect that algorithms which use these ideas—i.e., by reasoning about a limited set of logical facts, running a finite collection of linear-time traders, and using standard gradient descent techniques to find approximately reasonable prices—can likely achieve good performance in practice. This is an empirical conjecture, which remains to be tested.

References

Aaronson, Scott. 2013. "Why Philosophers Should Care About Computational Complexity." Chap. 10 in *Computability: Turing, Gödel, Church, and Beyond,* edited by B. Jack Copeland, Carl J. Posy, and Oron Shagrir, 261–328. Cambridge: MIT Press.

Campbell-Moore, Catrin. 2015. "How to Express Self-Referential Probability. A Kripkean Proposal." *The Review of Symbolic Logic* 8 (04): 680–704.

Carnap, Rudolf. 1962. *Logical Foundations of Probability.* Chicago, IL, USA: University of Chicago Press.

Christiano, Paul. 2014. *Non-Omniscience, Probabilistic Inference, and Metamathematics.* Technical report 2014–3. Berkeley, CA: Machine Intelligence Research Institute. `http://intelligence.org/files/Non-Omniscience.pdf`.

De Finetti, Bruno. 1979. *Theory of Probability.* New York, NY, USA: John Wiley & Sons.

Demski, Abram. 2012. "Logical Prior Probability." In *Artificial General Intelligence: 5th International Conference, AGI 2012,* 50–59. Lecture Notes in Artificial Intelligence 7716. New York: Springer.

Fagin, Ronald, Yoram Moses, Moshe Y. Vardi, and Joseph Y. Halpern. 1995. *Reasoning About Knowledge.* Cambridge: MIT Press.

Fallenstein, Benja, and Nate Soares. 2015. *Vingean Reflection: Reliable Reasoning for Self-Improving Agents.* Technical report 2015–2. Berkeley, CA: Machine Intelligence Research Institute. `https://intelligence.org/files/VingeanReflection.pdf`.

Gaifman, Haim. 1964. "Concerning Measures in First Order Calculi." *Israel Journal of Mathematics* 2 (1): 1–18.

Good, Irving John. 1950. *Probability and the Weighing of Evidence.* London: Charles Griffin.

Hacking, Ian. 1967. "Slightly More Realistic Personal Probability." *Philosophy of Science* 34 (4): 311–325.

Hay, Nicholas, Stuart J. Russell, Solomon Eyal Shimony, and David Tolpin. 2012. "Selecting Computations: Theory and Applications." In *Uncertainty in Artificial Intelligence (UAI-'12),* edited by Nando de Freitas and Kevin Murphy, 346–355.

Hintikka, Jaakko. 1962. *Knowledge and Belief.* Ithaca, N.Y.: Cornell University Press.

———. 1975. "Impossible Possible Worlds Vindicated." *Journal of Philosophical Logic* 4 (4): 475–484.

Hutter, Marcus, John W. Lloyd, Kee Siong Ng, and William T. B. Uther. 2013. "Probabilities on Sentences in an Expressive Logic." *Journal of Applied Logic* 11 (4): 386–420.

Kingma, Diederik P., and Max Welling. 2013. "Auto-Encoding Variational Bayes." arXiv: `1312.6114 [stat.ML]`.

Levin, Leonid A. 1974. "Laws of Information Conservation (Nongrowth) and Aspects of the Foundation of Probability Theory." *Problemy Peredachi Informatsii* (3): 30–35.

Li, Nan, William Cushing, Subbarao Kambhampati, and Sungwook Yoon. 2010. "Learning probabilistic hierarchical task networks to capture user preferences." *arXiv preprint arXiv:1006.0274.*

Pearl, Judea. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA: Morgan Kaufmann.

Ramaré, Olivier. 1995. "On šnirel'man's constant." *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze* 22 (4): 645–706.

Ramsey, Frank Plumpton. 1931. "Truth and Probability." In *The Foundations of Mathematics and other Logical Essays,* edited by Richard Bevan Braithwaite, 156–198. New York: Harcourt, Brace.

Rantala, Veikko. 1975. "Urn Models: A New Kind of Non-Standard Model for First-Order Logic." *Journal of Philosophical Logic* 4 (4): 455–474.

Russell, Stuart J. 2014. "Unifying Logic and Probability: A New Dawn for AI?" In *Information Processing and Management of Uncertainty in Knowledge-Based Systems: 15th International Conference IPMU 2014, Part I,* 442:10–14. Communications in Computer and Information Science, Part 1. Springer.

Savage, Leonard J. 1967. "Difficulties in the theory of personal probability." *Philosophy of Science* 34 (4): 305–310.

Solomonoff, Ray J. 1964. "A Formal Theory of Inductive Inference. Part I." *Information and Control* 7 (1): 1–22.

Von Neumann, John, and Oskar Morgenstern. 1944. *Theory of Games and Economic Behavior.* 1st ed. Princeton, NJ: Princeton University Press.