Supply and demand for logical uncertainty

Abhimanyu Pallavi Sudhir

23 April 2023

Abstract

Garrabrant induction assigns probabilities to logical sentences based on their prices in a prediction market. However, it does not discriminate based on the "importance" of a logical sentence – it does not provide a way to incentivize information about particular logical sentences of "interest" to us. We demonstrate that the mechanism can be made "incentive-compatible" with our interests with an appropriately-chosen automated market-maker, and suggest that the framework can be extended into more general models of bounded rationality and generally intelligent agents.

1 Introduction

A framework for working with logical uncertainty (uncertainty arising from not being logically omniscient) was defined in the seminal work of Garrabrant et al [1], which constructed an algorithm for assigning probabilities to logical sentences. The algorithm proceeds by constructing a market of logical sentences that pays off as a theorem enumerator outputs each sentence, where every polynomial-time trader (i.e. a trader for which there exists a polynomial p such that its strategy on each day n is computed in $p(n)$ time) participates in the market from some point based on some enumeration, and on each day n computing a rational approximation to the equilibrium price of the market by a brute-force enumeration of rational tuples. Naturally, this is a terribly inefficient algorithm – furthemore, it does not provide a way for us to incentivize the market to prioritize logical sentences of interest to us.

But this is hardly the only possible way to define a market: mechanism design and automated market-making are fields with extensive corpora of literature (the most widely-known of which is Logarithmic Market Scoring [2]), and although most work has focused on markets of perfectly rational agents, efficient outcomes can still be achieved by *program markets* – markets whose participants are programs – if they are incentive-compatible and have budget constraints, because smarter traders make more money and gain influence, while dumber ones go bankrupt (see [3–6] for discussion – we will state a formal version of this result for our uses in this paper).

We recast the Garrabrant algorithm with a logarithmic market maker and point out that this lets us prioritize logical sentences by appropriately setting the liquidity parameter. This requires an alternate definition of market optimality than the "inexploitability" condition satisfied by the Garrabrant algorithm, one which looks more like asymptotically optimizing a certain welfare function – and as a result there are many technical differences between our framework and standard Garrabrant induction:

- We do not (and cannot) require traders to be continuous functions – instead we allow markets that do not reach equilibrium (e.g. "the price of this asset will be ¡0.4 on day 5"), and leave it to traders and the market maker to not provide much liquidity to such markets.
- !!!! I'm not sure if this was the right call, I might have broken something big.!!! In the Garrabrant framework, propositional logic has a fixed exchange rate, so e.g. a stock in ϕ plus a stock in $\neg\phi$ can always be exchanged for \$1 – this shows up in the budget constraint (holding $\phi + \neg\phi$ allows you to exchange that for stocks worth \$1) and in the definition of exploitation (earning unbounded numbers of $\phi + \neg\phi$ stock counts as exploitation). This is true even if both ϕ and φ are unprovable, so unprovable statements still have non-zero price.

  [I think I did break something – what is the probability of a program halting? It's going to just be zero the way I've constructed it.]
- We accomodate any enumeration of traders (not necessarily polynomial-time) and account for the computational costs by deducting them from the traders' budgets.

It may be possible to extend our work to a framework where this welfare function is propagated from downstream tasks, i.e. itself based on a market mechanism. There has been speculation that program markets could be used as architectures of intelligent agents [7, 8], as they seem to possess the desired attributes of boundedly rational agents [9] – our work may be considered a step closer to the design of such agents, and to a general theory of bounded rationality.

2 Logarithmic logical inductor

Notation. The sets $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ mean what they always do, with $0 \notin \mathbb{N}$. Intervals, e.g. $[0, 1]$ will always be subsets of $\mathbb{Q}$ unless specified otherwise with a subscript. $f : A \to_\circ B$ is a function with finite support $\operatorname{supp} f$, and $\operatorname{finset} A = (A \to_\circ \{0, 1\})$, we use $f : (a : A) \to B$ as an alternative to lambda notation (and we will not bother to distinguish functions and dependent types), and the set of functions $A \to B$ may also be denoted as B^A. Types may be left implicit as __ when it is understood from context what sort of object is being referred to.

Constants. Let:

- $\mathcal{S}$ be a set of "sentences" in some language

- $\mathcal{T} : (t : \mathbb{N}) \to \text{finset}\,\mathcal{S}$ s.t. $s < t \implies \mathcal{T}(s) \subseteq \mathcal{T}(t)$ be an enumerator of "theorems"
- $\mathbb{P} = \mathcal{S} \to_\circ [0,1]$ is the "type of prices" (note in particular how $\mathcal{T}(t) \in \mathbb{P}$)
- $\mathbb{X} = \mathcal{S} \to_\circ \mathbb{Q}$ is the "type of portfolios"
- $\mathbb{X}' = \mathbb{Q} \times \mathbb{X}$ is the "type of portfolios including a cash term"
- Addition (+) between portfolios and cash amounts in the obvious way

Definition 2.1. A market-maker is a computable function $\mu : (t : \mathbb{N}) \to \mathcal{S} \to (\mathbb{Q}^2 \to \mathbb{Q}^2)$ – for given t, s, the function $\mu(t,s)(x_0, x_1)$ produces prices p_0, p_1 of "NO" and "YES" stocks for the sentence s given the inventory of those stocks x_0, x_1.

In particular for a logarithmic market-maker:

$$\mu(t,s)(x_0,x_1) = \frac{(\exp((-a_0 - x_0)/b), \exp((-a_1 - x_1)/b)}{\sum \exp((-a_i - x_i)/b)}$$

where (a_0, a_1) and b are the "bias" and "liquidity" of the market for sentence s at time t". $\sum_s b_{t,s} < \infty$ at every time.

Note how it may assign liquidity to an infinite number of sentences! The "initial price" posited by the market-maker for a sentence no one has traded on yet – the prior belief taken by the market-maker – is $\frac{(e^{-a_0/b}, e^{-a_1/b})}{e^{-a_0/b} + e^{-a_1/b}}$.

Definition 2.2. A trader is a computable function $\alpha : (t : \mathbb{N}) \to \mathbb{P}^t \to \mathcal{S} \to_\circ \mathbb{Q}$. $\alpha(t, \underline{p}, s)$ is the "fraction of its cash holdings exchanged for stock in s at time t based on price history $\underline{p}$". $\sum_s \alpha(t, \underline{p}, s) = 1$ but any individual $\alpha(t, \underline{p}, s)$ is allowed to be negative.

References

[1] Scott Garrabrant et al. "Logical Induction". In: *CoRR* abs/1609.03543 (2016). arXiv: `1609.03543`. URL: `http://arxiv.org/abs/1609.03543`.

[2] Robin Hanson. "Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation". In: *The Journal of Prediction Markets* 1.1 (Dec. 13, 2012), pp. 3–15. ISSN: 1750-676X, 1750-6751. DOI: `10.5750/jpm.v1i1.417`. URL: `http://www.bjll.org/index.php/jpm/article/view/417` (visited on 04/02/2023).

[3] Dhananjay K. Gode and Shyam Sunder. "Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality". In: *Journal of Political Economy* 101.1 (Feb. 1993). Publisher: University of Chicago Press, pp. 119–137. DOI: `10.1086/261868`. URL: `https://doi.org/10.1086/261868`.

Algorithm 1 Logarithmic logic market

```
1: for $\alpha$ do
2:     $x_\alpha \leftarrow \dots$                                                    ▷ initialize agent budgets
3: while $t < \infty$ do
4:     for $\alpha$ do
5:         $q \leftarrow \alpha(t, \underline{p})x_\alpha^{\text{cash}}$                              ▷ agent decides how to spend
6:         for $s \in \operatorname{supp} q$ do
7:             $x \leftarrow (\int \mu(t, s, x_{\text{mm},s} + x)\, dx)^{-1}(q_s)$          ▷ compute stock qty
8:             $x_\alpha^{\text{cash}} \leftarrow x_\alpha^{\text{cash}} - q_s$                              ▷ update portfolios
9:             $x_{\alpha,s} \leftarrow x_{\alpha,s} - x$
10:            $x_{\text{mm},s} \leftarrow x_{\text{mm},s} + x$
11:            $p_{t,s} = \mu(t, s, x_{\text{mm},s})$                                     ▷ update price
12:    for $s \in \mathcal{T}(t)$ do
13:        for $\alpha$ do:
14:            $x_{\alpha,\text{cash}} \leftarrow x_{\alpha,s}$                                    ▷ pay off stockholders
15:            $x_{\alpha,s} \leftarrow 0$
```

[4] D. K. Gode and S. Sunder. “What Makes Markets Allocationally Efficient?” In: *The Quarterly Journal of Economics* 112.2 (May 1997). Publisher: Oxford University Press (OUP), pp. 603–630. DOI: 10.1162/003355397555307. URL: https://doi.org/10.1162/003355397555307.

[5] Alan Schwartz. “How Much Irrationality Does the Market Permit?” In: *The Journal of Legal Studies* 37.1 (Jan. 2008). Publisher: University of Chicago Press, pp. 131–159. DOI: 10.1086/519963. URL: https://doi.org/10.1086/519963.

[6] Karim Jamal, Michael S. Maier, and Shyam Sunder. “Simple Agents, Intelligent Markets”. In: *SSRN Electronic Journal* (2015). Publisher: Elsevier BV. DOI: 10.2139/ssrn.2478665. URL: https://doi.org/10.2139/ssrn.2478665.

[7] Caspar Oesterheld. *Futarchy implements evidential decision theory*. The Universe from an Intentional Stance. Dec. 18, 2017. URL: https://casparoesterheld.com/2017/12/18/futarchy-implements-evidential-decision-theory/ (visited on 04/02/2023).

[8] Gwern Branwen. “Evolution as Backstop for Reinforcement Learning”. In: (Dec. 6, 2018). URL: https://gwern.net/backstop (visited on 04/07/2023).

[9] Caspar Oesterheld, Abram Demski, and Vincent Conitzer. “A theory of bounded inductive rationality”. In: 2021.