machine learning & knowledge extraction

Article

Hardness of Learning in Rich Environments and Some Consequences for Financial Markets

Ayan Bhattacharya [1,2]

[1] Zicklin School of Business, Baruch College, The City University of New York, New York, NY 10010, USA; Ayan.Bhattacharya@baruch.cuny.edu

[2] Booth School of Business, University of Chicago, Chicago, IL 60637, USA

Abstract: This paper examines the computational feasibility of the standard model of learning in economic theory. It is shown that the information update technique at the heart of this model is impossible to compute in all but the simplest scenarios. Specifically, using tools from theoretical machine learning, the paper first demonstrates that there is no polynomial implementation of the model unless the independence structure of variables in the data is publicly known. Next, it is shown that there cannot exist a polynomial algorithm to infer the independence structure; consequently, the overall learning problem does not have a polynomial implementation. Using the learning model when it is computationally infeasible carries risks, and some of these are explored in the latter part of the paper in the context of financial markets. Especially in rich, high-frequency environments, it implies discarding a lot of useful information, and this can lead to paradoxical outcomes in interactive game-theoretic situations. This is illustrated in a trading example where market prices can never reflect an informed trader's information, no matter how many rounds of trade. The paper provides new theoretical motivation for the use of bounded rationality models in the study of financial asset pricing—the bound on rationality arising from the computational hardness in learning.

Keywords: learning; computational hardness; bounded rationality; market efficiency; high-frequency trading; rich environment

Citation: Bhattacharya, A. Hardness of Learning in Rich Environments and Some Consequences for Financial Markets. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 467–480. https://doi.org/10.3390/make3020024

Academic Editor: Isaac Triguero

Received: 22 April 2021
Accepted: 24 May 2021
Published: 28 May 2021

1. Introduction

This paper studies the computational feasibility of the standard model of learning in economic theory. At the heart of this model is the Bayesian update formula, with no restriction on the format of the input data used in the update. Using hardness results from Bayesian Network theory, it is shown that the update algorithm does not have a polynomial implementation unless an independence condition holds in the data. However, there is no polynomial algorithm to check for such an independence condition, rendering the overall learning model computationally infeasible. Having established this result, the paper explores some paradoxical consequences of using the learning model in financial environments like high-frequency trading.

A rough intuition of the computational hardness result can be conveyed in a simple example. Suppose one has observed the outcomes from a large number of biased coin tosses and knows the probabilities $p(\text{Coin}_1 = H, \text{Coin}_2 = H) = 1/6$, $p(\text{Coin}_1 = H, \text{Coin}_2 = T) = 1/2$, $p(\text{Coin}_1 = T, \text{Coin}_2 = H) = 1/4$, $p(\text{Coin}_1 = T, \text{Coin}_2 = T) = 1/12$. Now, if one had to calculate $p(\text{Coin}_1 = H)$, one would need to sum over (or marginalize) the states of the second coin. That is, $p(\text{Coin}_1 = H) = p(\text{Coin}_1 = H, \text{Coin}_2 = H) + p(\text{Coin}_1 = H, \text{Coin}_2 = T) = 2/3$. If there were three coins, one would have to sum over $2 \times 2 = 4$ states of other coins. With four coins, it is a sum over $2 \times 2 \times 2 = 8$ states. With n coins, it is a sum over 2^n states. Thus, the number of additions that need to be performed is an exponential computation. By its very nature, an exponential explodes quickly, making the computation infeasible in all but the simplest scenarios.

Of course, there could be shortcuts to this computation. If the coin tosses were independent, one could bypass such a calculation completely. Suppose one had observations from three independent coin tosses that recorded equal probability for all possibilities; in that case, the calculation of $p(\text{Coin}_1 = H)$ would simply be $p(\text{Coin}_1 = H) = \frac{p(\text{Coin}_1=H,\text{Coin}_2=H,\text{Coin}_3=H)}{p(\text{Coin}_1=H,\text{Coin}_2=H,\text{Coin}_3=H) + p(\text{Coin}_1=T,\text{Coin}_2=H,\text{Coin}_3=H)} = \frac{1/8}{1/8 + 1/8} = 1/2$. This calculation involves only a single addition and division. No matter how many coins one had, the structure of this calculation stays the same due to independence; thus, it is always computationally feasible. In fact, there is nothing special about choosing the state of the other coins (i.e., Coin_2 and Coin_3) as heads in the calculation. One could equally well choose their states to be tails—or for that matter, any consistent combination of heads and tails—because of the independence condition.

For the information update computation in the economic learning model to be feasible, a similar independence condition must hold in the data. Thus, the focus of the analysis in the paper is on checking the computational feasibility of finding this independence condition. It is shown that deducing such a condition from the data is computationally infeasible; specifically, the problem belongs to the class NP-hard. To establish this result, the paper builds on tools from the Graphical models area in theoretical machine learning. In summary, the results show that the standard learning model in economic theory becomes computationally infeasible in realistic settings like financial markets.

Why do we not usually run into such computational infeasibility issues in the prevalent models in financial economics? For instance, the vast literature on financial market trading spawned by the works of Grossman and Stiglitz [1], Glosten and Milgrom [2] and Kyle [3] is, in a certain sense, an exploration of the implications of the economic learning model in complicated trading scenarios, and one rarely comes up against computational infeasibility in such models. This is because the models make sprawling common knowledge assumptions. For example, it is common knowledge that there is an informed trader in the market, it is common knowledge that the noise order flow is independent of asset value, and so on. However, such knowledge has to be obtained from data in the real world, and this is where the computational infeasibility lies. In the example above with coins, if we were "given" the independence of the tosses, calculating the probability is a simple matter. The hardness lies in deducing such an independence assumption from the data.

The latter part of the paper explores some counterintuitive implications of holding on to the learning model in economic settings where it should be computationally infeasible. It is shown that the risks are higher in rich environments with a dense number of inputs, for instance, a high-frequency trading environment. This is because the richer the environment, the more the information that needs to be discarded to use the model. Discarding more information increases the chances that a decision maker's subjective model of reality deviates substantially from true reality, leading to paradoxes in interactive situations. The paper provides a pathological example that exploits this feature. In a fairly standard Glosten and Milgrom [2] setting, it is shown that prices may never reflect information, implying that markets may stay permanently inefficient.

The rest of the paper is organized as follows. Section 2 provides a literature review. Section 3 describes the model and establishes the hardness results. Section 4 defines a rich environment and explores the consequences of the hardness of the learning computation in such environments. The pathological example illustrating the lack of information percolation in trading is in Section 4.3. Section 5 concludes. The proofs are included in the text when they add to the narrative; otherwise, they can be found in Appendix A. Appendix A also provides a brief overview of the theory of directed acyclic graphs.

2. Literature Review

This paper is part of a recent literature that extends tools from theoretical machine learning to elucidate issues in econometrics and economic learning theory (Athey and Wagner [4], Chernozhukov et al. [5], Dao et al. [6], Iskhakov et al. [7], Oprescu et al. [8], Singh et al. [9], Sverdrup et al. [10], Syrgkanis et al. [11], Syrgkanis and Zampetakis [12]).

Within machine learning theory, the paper builds on work in the area of Bayesian networks (Koller and Friedman [13], Pearl [14]). In particular, the paper extends the large sample computational hardness results in Chickering et al. [15] to a setting that is important in economic theory. A number of papers have proposed algorithms for finding Bayesian network structures despite the computational hardness of the problem (Caravagna and Ramazzotti [16], Constantinou et al. [17], Malone et al. [18], Platas-Lopez et al. [19], Talvitie et al. [20], Zhang et al. [21]; and Scarnagatta et al. [22] for a survey of the older literature). Unlike these papers, the focus here is on showing that hardness affects a large class of learning problems in economic settings, and that can lead to counterintuitive results in applied areas like finance.

The paper also contributes to the bounded rationality literature in economic theory (Rubinstein [23], Spiegler [24]). This literature studies the consequences of sub-optimal decision making in interactive situations (Ellis and Piccione [25], Esponda and Puozo [26], Eyster et al. [27], Eyster and Piccione [28], Jehiel [29], Jehiel [30], Jehiel and Koessler [31], Mailath and Samuelson [32], Spiegler [33], Steiner and Stewart [34]). All of these papers take bounded rationality as the starting point of their analysis—without investigating the cause for boundedness. One way to think about the hardness result in this paper is that it provides a rigorous theoretical justification for the assumption of bounded rationality in such economic models. Specifically, the bound on rationality in our setting comes from the computational hardness of the learning model. In areas like financial market microstructure that use learning models extensively, systematic use of computationally bounded rationality has the potential to uncover many new features of modern algorithm driven asset pricing.

3. Hardness of Learning

This section first describes the modeling framework of the paper. It then establishes that learning, as defined in standard models in financial economics, is a computationally hard problem.

3.1. Modeling Framework

Let $X = X_1 \times \cdots \times X_m$ be a finite set of states of the world, with $m \geq 2$. The set X_i represents the set of values the variable x_i may take, and the cardinality of each X_i is at least two. $M = \{1, \ldots, m\}$ is the set of variable labels. To reduce the notational burden, the variables are often referred to just by their labels. The variables may be interpreted as recording the outcomes from an incomplete information game. Thus, for example, labels $N \subseteq M$ could record the actions of the players, label $s \in M$ could record the state of nature, labels $T \subseteq M$ could record the private signals available to some subset of the players about the state of nature, while labels $R \subseteq M$ could represent other outcomes recorded in the game (for instance, payoffs to players, or price in the case of financial market games). The symbol $\subseteq$ is used to denote a "strict subset or equality", while the symbol $\subset$ is reserved for a strict subset.

Let $p \in \Delta(X)$ be a probability distribution over the states of the world. That is, for every potential combination of variable values in X, p gives a probability. Like Spiegler [33], it is assumed here that p is representable as a directed acyclic graph over X. Technically, this means that p is perfect with respect to a directed acyclic graph. The typical interpretation (Spiegler [33]) is that there is a historical database consisting of (infinitely) many joint observations of the variables, and this gives the probability distribution p. Unless otherwise specified, the maintained assumption will be that the outcomes recorded in the historical database are optimal. This is a departure from Spiegler [33], where the focus is on subjective directed acyclic graphs used by decision makers. The assumption is without loss of generality in our setup. (See Appendix A for a discussion of the general case).

The focus will be on the learning problem of a single *uninformed player* (call UP) in this environment. The UP ("she") has access to only a limited set of variables in the historical database; in other words, only a strict subset $O \subset M$ is observable to the UP. This models the fact that in realistic situations, such as financial markets, uninformed players do not know the value of many variables—for instance the payoffs to the other players, or some of the private signals available to other players—even after the conclusion of the game. For simplicity, it will be assumed that a single variable is unobservable to the UP. The probability distribution from historical records available to the UP, denoted by p_O, is thus a marginalization of p over the set O. In this setting, the question studied is the following: Given the value of a variable x_β in UP's observable universe, how does the UP infer the value of another variable x_α, in her observable universe? That is, given $\alpha, \beta \in O$, how does the UP calculate $p_O\big(x_\alpha | x_\beta\big)$? In the sequel, for convenience, x_α will be written as α, and x_β as β. Then, the UP's problem could be framed as follows:

Problem 1 (UP's problem)**.** *Given p_O over a set $O \subset M$, obtain*

$$p_O(\alpha|\beta), \;\; \alpha, \beta \in O. \tag{1}$$

A learning problem like this is at the heart of numerous applications of Bayesian models to financial markets. For instance, a typical market making algorithm in a financial market has access to a large historical database of trades and wants to infer the best price to quote, given the value of an observable variable like order-flow size.

3.2. Establishing Hardness Results

The traditional approach to solving the UP's learning problem would be to use the Bayesian update formula

$$p_O(\alpha|\beta) = \frac{p_O(\alpha, \beta)}{p_O(\beta)}. \tag{2}$$

However, a moment's reflection makes it clear that the calculations involved are not trivial given a joint probability distribution over $O \subset M$. If every variable were potentially correlated with every other variable in M, Equation (2) would entail dividing two marginal probabilities, each the result of summation over an exponentially large number of variable combinations. This follows from the way marginals are calculated: in order to obtain the marginal probability of a random variable, one needs to sum the joint probability values over all possible states of other random variables. The following proposition encapsulates this observation.

Let us use the term *simple Bayesian inference* for Bayesian update under the assumption that every variable is potentially correlated with every other variable in set M.

Proposition 1. *In simple Bayesian inference, the number of elementary operations grows exponentially with the number of observable variables.*

Proof. Given a joint probability distribution over the set $O \subset M$, the denominator in Equation (2) becomes

$$p_O(\beta) = \sum p_O\Big((x_i)_{i \in O, i \neq \beta}, \beta\Big). \tag{3}$$

The term on the RHS has an exponential number of terms: if the number of states for a variable is at least two, the sum has at least $2^{|O|-1}$ terms. A similar logic holds for the numerator in Equation (2). We, therefore, have the result in the proposition. □

In other words, the simple Bayesian algorithm renders the inference problem exponentially hard. Does this mean that problem (1) is out of bounds for the UP using the Bayesian update formula? Obviously not, because every variable need not be correlated with every other variable in M. What we need is a general condition on the correlation structure of variables that constrains the number of operations in the Bayesian update formula (2) to grow at best at a polynomial rate. Once we have such a condition, we will have to ascertain if it can be checked by the UP in polynomial time. The following example illustrates the basic idea.

Example 1. *Consider a set of three observable variables x_1, x_2 and x_3. The first variable takes one of two values: heads or tails; that is $X_1 = \{h, t\}$. The second variable takes one of two values: up or down; that is $X_2 = \{u, d\}$. Finally, the third variable takes one of two values: left or right; that is $X_3 = \{l, r\}$. The joint probability distribution is given by the following eight tuples*

$$\begin{aligned} &p_O(h,u,l) = 1/18;\ p_O(h,u,r) = 1/9;\ p_O(h,d,l) = 1/6;\ p_O(h,d,r) = 1/3; \\ &p_O(t,u,l) = 1/12;\ p_O(t,u,r) = 1/6;\ p_O(t,d,l) = 1/36;\ p_O(t,d,r) = 1/18. \end{aligned} \tag{4}$$

Suppose our UP is tasked with calculating $p_O(h|u)$. The simple Bayesian algorithm would dictate that she calculate

$$p_O(h,u) = 1/18 + 1/9, \tag{5}$$
$$p_O(u) = 1/18 + 1/9 + 1/12 + 1/6, \tag{6}$$

to obtain $p(h|u) = p_O(h,u)/p_O(u) = 2/5$. Thus, two terms are summed in the numerator, four in the denominator, followed by a division. However, notice that the same result can be obtained by staying completely indifferent to the state of the third variable. That is, taking the ratio of

$$p_O(h,u,l) = 1/18, \tag{7}$$
$$p_O(u,l) = 1/18 + 1/12, \tag{8}$$

giving $p_O(h,u,l|u,l) = p_O(h|u) = 2/5$. Similarly,

$$p_O(h,u,r) = 1/9, \tag{9}$$
$$p_O(u,r) = 1/9 + 1/6, \tag{10}$$

giving $p_O(h,u,r|u,r) = p_O(h,u,l|u,l) = p_O(h|u) = 2/5$. Calculating $p_O(h,u,l|u,l)$ or $p_O(h,u,r|u,r)$ involves summing just two terms in the denominator followed by a division, a substantial reduction in the number of operations over simple Bayesian inference. What feature of the problem makes this possible? A quick check reveals that x_1 is independent of x_3 given x_2, and similarly x_2 is independent of x_3 given x_1. It is this conditional independence that allows the reduction in the number of operations.

Can we generalize the idea in the example? In other words, can we come up with a condition for the correlation structure in the data that can make the Bayesian computation polynomial and show that such a condition is unique? Indeed we can, and this result forms the content of the next proposition. The basic logic is fairly simple. If all the variables involved in the inference problem were conditionally independent of additional variables beyond a point, then we could simply ignore the multiple states of these additional variables, choosing any single state for the inference calculation. In the example above, we obtained the same conditional probability irrespective of whether we assumed x_3 took the value l or r. This means that the number of operations needed for the inference calculation does not increase with these additional variables. Due to the mathematical form of the joint probability distribution, such a condition is unique.

Recall the notation for variable labels. An uppercase letter is used to denote the set of variable labels and the corresponding lowercase letter to represent the cardinality of the set. Thus, $D = \{1, \ldots, d\}$ represents a set of variable labels and d denotes the cardinality of this set. Any variable belonging to this set would be denoted by $(x_k)_{k \in D}$. For any two variables x_i and x_j, i and $j \notin D$, the notation $x_i \perp_{p_O} x_j | D$ means that x_i is independent of x_j under p_O conditional on knowledge of the variables $(x_k)_{k \in D}$. This form of independence is also called *conditional independence*.

Proposition 2 (d-bound condition)**.** *The number of elementary operations in the Bayesian update formula is polynomial if and only if there exists a bound d, such that α and β are independent of all additional variables beyond the bound, given the values of a subset of variables in D.*

In other words, if and only if there exists a $d = \max(d_\alpha, d_\beta)$ *and set* $D = D_\alpha \cup D_\beta$*, such that* $\alpha \perp_{p_O} x_i | K_\alpha \subseteq D_\alpha$*,* $\beta \in K_\alpha$*, for all* $(x_i)_{i \notin D_\alpha}$*; and* $\beta \perp_{p_O} x_i | K_\beta \subseteq D_\beta$*, for all* $(x_i)_{i \notin D_\beta}$*.*

Proof. Assume the conditional independence conditions hold for α and β. Let $K = K_\alpha \cup K_\beta$. Choose $x_i \notin D_\alpha \cup D_\beta$. Then, by the definition of conditional independence,

$$p_O(\beta | K) = p_O(\beta | K, x_i). \tag{11}$$

This means that

$$\sum_{O \backslash K \cup \{\beta\}} p_O\big(\beta, (x_j)_{j \in K}\big) = p_O(x_i) \sum_{O \backslash K \cup \{i, \beta\}} p_O\big(\beta, (x_j)_{j \in K}, x_i\big). \tag{12}$$

Similarly, since α and β are independent of x_i conditional on knowing $(x_j)_{j \in K}$, we have that

$$\sum_{O \backslash K \cup \{\alpha, \beta\}} p_O\big(\alpha, \beta, (x_j)_{j \in K}\big) = p_O(x_i) \sum_{O \backslash K \cup \{i, \alpha, \beta\}} p_O\big(\alpha, \beta, (x_j)_{j \in K}, x_i\big). \tag{13}$$

Thus, in the Bayesian update formula (2), the common factor $p_O(x_i)$ in Equations (12) and (13) gets canceled when taking the ratio of $p_O(\alpha, \beta)$ and $p_O(\beta)$. Therefore, we can choose any element of X_i as x_i's realized state without loss of generality, and then use the joint probability distribution values corresponding to that state of x_i for our computation of $p_O(\beta | K)$. In effect, all x_i for $i > d$ become single state variables for the computation of the Bayesian update formula. Thus, the number of operations is now bounded and, thus, polynomial in the number of variables.

For the other direction, let us first write out $p_O(\beta)$ in terms of the joint probability distribution, that is

$$p_O(\beta) = \sum_{O \backslash \{\beta\}} p_O\Big(\beta, (x_j)_{j \in O \backslash \{\beta\}}\Big). \tag{14}$$

It is easy to see that the number of terms in the summation increases exponentially whenever the cardinality of X_j is greater than or equal to two. The only way that this summation is rendered non-exponential—and polynomial—is if the effective cardinality of X_j somehow becomes one for all j greater than some bound d_β; that is, if

$$p_O(\beta | K_\beta) = p_O(\beta | K_\beta, x_j), \tag{15}$$

for some $K_\beta \subseteq D_\beta$. Equation (15) is the definition of conditional independence for β. A similar argument for $p_O(\alpha | \beta)$ gives us the conditional independence condition for α. This completes the proof of the result. □

Let us call the condition in Proposition 2 the *d-bound condition.* That is to say, the number of elementary operations in the Bayesian update formula is polynomial if and only if the correlation structure of variables in M satisfy the d-bound condition for α and β. The next step on the agenda is to gauge how long it takes to check the d-bound condition. If the check is polynomially easy, then the Bayesian update formula is still useful for the UP. Specifically, the UP would first check the d-bound condition in the data, and if it holds, apply the Bayesian update formula to solve her problem. However, if the check turns out to be hard, then the Bayesian update formula is no longer a realistic tool for the UP.

The following theorem shows that checking the d-bound condition is NP-hard. The theorem builds on hardness results from Probabilistic graphical models. Chickering et al. [15] showed that checking the existence of a directed acyclic graph with a bounded number of parameters is NP-hard when a decision maker uses a joint probability distribution over a subset of variables (Appendix A defines directed acyclic graphs and related terms). It is shown below that checking the d-bound condition is, in a sense, equivalent to checking the condition in Chickering et al. [15]. Thus, it is also hard.

Theorem 1. *Given the variable set M, there does not exist a polynomial algorithm for a UP to check the d-bound conditions for α and β.*

Proof. Let me describe the main intuition of the proof before providing a more formal argument. Appendix A provides a brief overview of concepts from directed acyclic graphs (DAG) that are used in the proof.

First, note that knowing Pa_j, De_j and Nd_j for all $j \in M$, is equivalent to knowing the Bayesian network DAG. In other words, knowing the set of parents, descendants and nondescendants for every variable immediately gives the DAG that corresponds to the assignments. Next, observe that if the d-bound condition were satisfied for α and β, it would imply knowing D_α, K_α, D_β and K_β. Since $Pa_\alpha = K_\alpha$, $De_\alpha = D_\alpha \backslash K_\alpha$, and $Nd_\alpha = M \backslash D_\alpha$, if there was a polynomial algorithm that told the UP that the d-bound condition was satisfied, it would mean that that they would know the parent, descendant and nondescendant set for α and β in polynomial time. Now, α and β are arbitrarily chosen variables in the observable set; thus, such an algorithm could be used to obtain Pa_j, De_j and Nd_j for any $j \in O$; that is, any variable in the observable set. This, in turn, would immediately give the parent, descendant and nondescendant set for the unobservable variable. In effect, one would know the parent, descendant and nondescendant sets for all variables, and thus the Bayesian network DAG, and this would contradict Proposition A1 in Appendix A below, a version of Chickering et al. [15].

This chain of arguments can be stated more formally in two steps.

Step 1: Denote the variable unobservable to the UP by u. Now, for each $i \in O$, u has to belong to either Pa_i, De_i or Nd_i. Thus, knowing the assignment of Pa_i, De_i and Nd_i for all $i \in O$ implies knowing Pa_u, De_u and Nd_u.

Step 2: Given a bound d, suppose there were a polynomial algorithm to check the existence of Pa_i, De_i and Nd_i for $i \in O$. Such an algorithm would output a yes or no answer in polynomial time; that is, we would have $a_i \in \{y_i, n_i\}$ for each $i \in O$ in polynomial time. Since the sum of polynomials is also a polynomial, this implies that in polynomial time we would have the tuple $(a_i)_{i \in O}$. If $a_i = y_i$ for all $i \in O$, then from Step 1, we would have $a_u = y_u$ for the unobservable variable, too. We would thus assert the existence of a Bayesian network DAG for bound d. On the other hand, if $a_i = n_i$ for any $i \in O$, we would immediately assert the non-existence of a DAG for the bound. Either way, we would reach a decision in polynomial time, contradicting Proposition A1 in the appendix. This completes the proof. □

Thus far, we have assumed that the joint probability distribution p is representable as a directed acyclic graph. In Appendix A, it is shown that a similar logic goes through even when p might not be representable in this manner. Theorem 1 and Propositions 1, 2 and A3 imply that no matter what, the Bayesian update formula does not have a polynomial implementation for our problem. This gives the following corollary.

Corollary 1. (Theorem 1 and Proposition A3). *For any two variables α and β in the observable set $O \subset M$, given a joint probability distribution p_O over O, there does not exist a polynomial implementation for the Bayesian update formula. In other words, the UP cannot solve her problem in polynomial time with the Bayesian update formula.*

4. Some Consequences of Hardness

This section explores the consequences of the computational hardness of the learning model described previously. I first clarify the meaning of a rich environment, then show that the risks from using the learning model are high in rich environments. Finally, an illustrative example is provided that demonstrates these risks in a market trading setting.

4.1. Rich Environments

An environment is termed *rich* if it can separate a hard solution to a problem from an easy solution (the term 'hard' is used here for non-polynomial solution procedures). If not, the environment is *sparse*. The separating variable is the time needed for computation. In a rich environment, a hard solution takes an impossibly high amount of time to solve, thus rendering it for all practical purposes infeasible.

The general relation is shown in Table 1. A polynomial or easy solution for a problem is feasible irrespective of whether the environment is sparse or rich. A hard solution, however, is computationally feasible only in sparse environments. This is because a hard solution to a problem often involves an exhaustive search of the solution space. This might be feasible in sparse environments, but in rich environments, it is practically impossible. Thus, the hardness of Bayesian learning matters in rich environments, but in sparse environments, not so much. In the next section, a related but different perspective on the sparse versus rich divide is provided—in terms of the information that gets discarded when one insists on using the Bayesian update.

Table 1. Computational feasibility in sparse and rich environments. *CF* = Computationally Feasible, *CI* = Computationally Infeasible.

	Sparse Environment	**Rich Environment**
Easy Solution	*CF*	*CF*
Hard Solution	*CF*	*CI*

4.2. Coping with Hardness

Given that there does not exist a polynomial implementation for Bayesian update, how would the UP learn in rich environments? Essentially, what we are asking for is realistic approaches to the NP or exponentially hard class of problems. There is no general solution that works everywhere, and the approaches are tailored to the specific problem at hand. For instance, if the environment is sparse, it might just be feasible to parse through the solution space exhaustively. In economics, many of these approaches are studied under the rubric of bounded rationality. In computer science, such approaches come under approximation algorithms (Williamson and Shmoys [35]).

In our setup, the UP has two broad ways to solve her problem in rich environments. The first approach could be to abandon the Bayesian update formula altogether. Any number of heuristics could replace Bayesian updating, depending on the context. The second approach could be to hold on to the Bayesian update formula but impose a d-bound such that the update becomes feasible in polynomial time, irrespective of whether or not the condition actually holds in the data. Once again, a number of techniques could aid the UP in deciding how to choose her d-bound. When economists use the Bayesian update formula to model learning in rich environments, they are implicitly assuming the second approach in their setting. In the sequel, some consequences of this second approach are explored—without making any assumptions about the technique(s) used to select the d-bound. Papers like Ellis and Piccione [25], Esponda [36] and Spiegler [33] provide behavioral approaches to model such situations.

Intuitively, the d-bound means that beyond a certain point, the UP assumes all further variables to be independent of her variable of interest. This allows her to concentrate precious computational resources on the variables she deems important, thus achieving a possibly polynomial time Bayesian update. However, this involves an inherent tradeoff. The lower the time for the update computation, the greater the risk that useful information gets missed. One can show this rigorously in the present setting.

Recall the standard epistemic definition of information and knowledge (Aumann [37]). Given a finite set of states of the world X, a decision maker's *information* is a finite partition of X, represented by $\mathcal{F}$. The interpretation is that if the true state of the world is ω^*, the decision maker knows only the partition in $\mathcal{F}$ that contains ω^*. The decision maker *knows an event* $A \subseteq Y$ if and only if $\mathcal{F}(\omega^*) \subseteq A$. In this case, he knows the event A even though he may not know the true state of the world. A decision maker has more information if his partition allows him to know more events.

Example 2. *Suppose $X = \{\omega_1, \omega_2, \omega_3, \omega_4\}$. Consider the partition $\mathcal{F}_i = \{\{\omega_1, \omega_2\}, \{\omega_3\}, \{\omega_4\}\}$. If the true state of the world is $\omega^* = \omega_1$, with information $\mathcal{F}_i$, a decision maker only knows that the state of the world is either ω_1 or ω_2. So given an event $A = \{\omega_1, \omega_2\}$, he knows event A in the states ω_1 and ω_2, even though he cannot distinguish between ω_1 and ω_2. This is because $\mathcal{F}_i(\omega_1) = \mathcal{F}_i(\omega_2) = \{\omega_1, \omega_2\} \subseteq A$. Now suppose, instead, the decision maker had information $\mathcal{F}_j = \{\{\omega_1, \omega_2, \omega_3\}, \{\omega_4\}\}$. Can he then perceive event A? No, because $\mathcal{F}_j(\omega_1) = \mathcal{F}_j(\omega_2) = \{\omega_1, \omega_2, \omega_3\} \not\subseteq A$. In other words, partition $\mathcal{F}_j$ has less information than partition $\mathcal{F}_i$.*

The intuition in the example is quite general. Since it helps in the proof of the next proposition, let us state it as a Lemma.

Lemma 1. *A coarser partition of the states of the world implies lesser information. In other words, if*

$$\begin{aligned} &\mathcal{F}_j(\omega) \supseteq \mathcal{F}_i(\omega) \text{ for all } \omega \in X, \\ \text{and } &\mathcal{F}_j(\theta) \supset \mathcal{F}_i(\theta) \text{ for some } \theta \in X, \end{aligned} \tag{16}$$

then, using $\mathcal{F}_j$, a decision maker knows fewer events than using $\mathcal{F}_i$.

Proof. For any event A, if $\mathcal{F}_j(\omega) \subseteq A$, then $\mathcal{F}_i(\omega) \subseteq A$ from condition (16). Thus, if $\mathcal{F}_j$ implies an event is known, so does $\mathcal{F}_i$. However, since $\mathcal{F}_j(\theta) \supset \mathcal{F}_i(\theta)$, there is at least one element in $\mathcal{F}_j(\theta)$ that is not in $\mathcal{F}_i(\theta)$. Denote such an element by θ^*. Then, using $\mathcal{F}_i$, the event $\{\theta^*\}$ is known but using $\mathcal{F}_j$ it is not. □

Though there are more general ways to define the notion of relative coarseness, in the proposition we have used the simplest definition that serves our purpose: partition $\mathcal{F}_j$ is *coarser* than $\mathcal{F}_i$ if $\mathcal{F}_j(\omega) \supseteq \mathcal{F}_i(\omega)$ for all $\omega \in X$, and $\mathcal{F}_j(\theta) \supset \mathcal{F}_i(\theta)$ for some $\theta \in X$.

We are now ready to prove the main result of this section. The following proposition shows that if the UP chooses a lower d-bound to speed up her computation, she discards more information. The intuition behind the result is not hard to see. A lower bound implies that the UP assumes more variables to be independent of her problem. This allows her to concentrate resources on a lesser number of variables, leading to faster computation. However, assuming more variables to be independent of the problem also implies discarding more information. A useful interpretation of this result, in terms of the richness of the environment, is provided following the proposition.

Proposition 3. *If the UP chooses a lower d-bound for the Bayesian update, she speeds up her computation but has less information. In other words, holding computational resources constant, the speed of computation and amount of information are inversely related when using the Bayesian update formula for learning.*

Proof. Let d_1 and d_2 be two potential d-bounds with $d_1 < d_2$. For the Bayesian update formula, the states of the world that the bound d_1 distinguishes are $Y_1 = X_1 \times \cdots \times X_{d_1}$, while the states that d_2 distinguishes are $Y_2 = X_1 \times \cdots \times X_{d_1} \cdots \times X_{d_2}$. Thus, a coarser partition with d_1, and from Lemma 1, less information.

Next, since the UP knows only the joint probability distributions from the historical database, she needs to sum over states to calculate probabilities of events. (To keep notation simple, I drop the subscript O for the probability distribution because it plays no part in the arguments of this proof.) In other words, given a joint probability distribution p, she calculates

$$p(\beta) = \sum_M p\Big(\beta, (x_j)_{j \in M \setminus \{\beta\}}\Big). \tag{17}$$

As discussed in the proof of Proposition 2, if a subset of the variables in M, say $E \subset M$ is assumed independent of β, the UP does not need to sum over states for such variables. She can assume any arbitrary state as the realization for variables in E, and as long as they are consistent throughout, it does not affect the result of their Bayesian update calculation (refer to Equations (12) and (13)). Thus, the larger the set E, the faster her computation. Since $E_1 = \{d_1 + 1, \ldots M\} \supset \{d_2 + 1, \ldots M\} = E_2$, we have the result. □

Discarding more information increases the risk of deviating from reality. Thus, in essence, Proposition 3 says that there is a higher likelihood of UP's subjective model of reality deviating from objective reality as she speeds up her Bayesian update computation. This gives another interpretation of the sparse and rich environment difference. In a sparse environment, since the number of variables is limited to begin with, only a little speeding up might be enough to make the Bayesian update calculation feasible. In a rich environment, on the other hand, the speeding up required for feasibility is large. Thus, the subjective models of agents in rich environments might deviate much more substantially from objective reality. The next section provides a concrete example of such a deviation and explores the consequences.

4.3. Illustrative Example: Information Percolation in Trading

A canonical principle in financial market theory is that any trading interaction between an informed trader and an uniformed trader leads to some information percolation (Glosten and Milgrom [2], Grossman and Stiglitz [1], Kyle [3]). However, when learning is hard, this principle is violated. To show this in a concrete setting, we need some elementary notions from combinatorics theory.

Consider the following succession of numbers

$$(0, 0, 0, 1, 1, 1, 0, 1)^{\infty}. \tag{18}$$

The exponent ∞ means that the sequence in the bracket is repeated ad infinitum. This succession of numbers has two special properties: (i) given a sequence of any three consecutive numbers, the next number in the succession is unique, but (ii) given a consecutive sequence of less than three numbers, the next number is equally likely to be a 0 or 1. Such a succession is called a DeBruijn sequence, and such sequences are well-studied objects in combinatorics. (Ralston [38] provides an introduction to DeBruijn sequences. For an application to modeling bounded rationality, refer to Piccione and Rubinstein [39].) Specifically, the DeBruijn sequence in (18) has *order* 3 (the minimum size of the consecutive sequence such that the next number in the succession is predictable), and the *symbol set* (set of values that each number may take) is $\{0, 1\}$. A well-known result in this area states that for any finite order and symbol set, there exists a DeBruijn sequence (see Ralston [38]).

Consider an asset whose value v depends on three market variables (say, for concreteness: growth g, regulation r, and supply s). Each of these variables could be either 1 (high) or 0 (low), and their value is publicly known in the markets. Given the value of these market variables, the asset value is determined by the DeBruijn sequence in (18). Thus $g = 0, r = 0, s = 0 \Rightarrow v = 1$; $g = 0, r = 0, s = 1 \Rightarrow v = 1$; $g = 0, r = 1, s = 1 \Rightarrow v = 1$; and so on. There is an informed trader $\mathcal{I}$ in the market who knows the DeBruijn sequence. Thus, $\mathcal{I}$'s information consists of the succession $(0,\ 0,\ 0,\ 1,\ 1,\ 1,\ 0,\ 1)$, and the asset value is completely predictable to him. $\mathcal{I}$ buys when the value is high (say $\mathcal{I}$ = 1), and sells when value is low ($\mathcal{I} = 0$) . There is also an uninformed market maker $\mathcal{U}$ in this market who (in order to keep learning time under control) uses a d-bound of 2 (for concreteness, say he assumes that the asset value is independent of growth). The uninformed's prior is that the value is equally likely to be high or low. The trading mechanism is simple: $\mathcal{U}$ quotes a bid and ask price (to which he commits), and then $\mathcal{I}$ decides to buy or sell.

The traditional solution to this problem is the famous result from Glosten and Milgrom [2]: $\mathcal{U}$ quotes a bid of 0 and asks of 1, and there is no trade unless there are noise traders in the market to circumvent the no-trade theorem (Milgrom and Stokey [40]). Thus, a bid-ask spread results as a consequence of asymmetric information. However, there is an unanswered question that lurks in the background. How does the uninformed market maker label his counter-party as informed? The presumable way, in the real world, would be to check in the historical database. In this database, $\mathcal{U}$ would find that $r = 0$, $s = 0 \Rightarrow \mathcal{I}$ equals 0 or 1; $r = 0$, $s = 1 \Rightarrow \mathcal{I}$ equals 0 or 1; and $r = 1$, $s = 0 \Rightarrow \mathcal{I}$ equals 0 or 1. Since $\mathcal{U}$ uses a d-bound of 2, the value of g would not enter into his calculation at all. In fact, because of the property of DeBruijn sequences, $\mathcal{U}$ would deduce that $\mathcal{I}$ is equally likely to be a buy or sell in each case. In other words, behavior that is identical to a noise trader's. Thus, in this setting, $\mathcal{U}$ would fail to label an informed trader. To him, $\mathcal{I}$ is a noise trader. Thus, we have the following counterintuitive result.

Proposition 4. *In the setting described above, the uninformed trader sets both bid and ask price at* 0.5*, and the informed trader can trade repeatedly at this price without any information percolation.*

Proof. Follows from the discussion above and property of DeBruijn sequences. □

A complementary way to interpret this result would be to think about the informational efficiency of this market. Proposition 4 says that the information about the DeBruijn sequence is never reflected in prices.

Corollary 2 (Proposition 4)**.** *There exists information that is never reflected in market prices.*

Though, for concreteness, we have worked with a specific DeBruijn sequence, this is a fairly general result. This is because, as described above, DeBruijn sequences exist for any finite order and symbol set. So, no matter how large $\mathcal{U}$ makes his d-bound, there will always be some information that will escape him, thus never enter prices.

5. Conclusions

It seems intuitively apparent that routine tasks must become progressively harder as the environment in which they are performed becomes complex. The paper makes this intuition precise in the context of the standard learning model in economic theory, and shows that the tasks involved in learning become computationally infeasible in the rich environment of modern financial markets. Continuing to use the standard model of learning, despite computational infeasibility, indicates deviation of an agent's subjective model from objective reality.

This insight has many implications, not just for financial asset pricing and market efficiency but also for interactive machine learning. When multiple machine learning algorithms with distinct subjective versions of reality interact, how far can the interactive epistemics deviate from objective reality? This is a problem that financial markets have to face everyday, given most trading firms today run machine learning algorithms to manage their market trading. In the paper, we saw a pathological example of trading where no information at all was conveyed in the prices. It would be interesting to extend such results systematically to develop a general theory of bounded, interactive machine learning.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A

A directed acyclic graph (DAG) is defined as a set of nodes (vertices) and directed edges that has no directed paths starting and ending at the same node (i.e., no cycles). DAGs are an important tool in machine learning to study causal reasoning, and the broad technique goes by the name Bayesian networks. It was pioneered by Judea Pearl and others (Pearl [14], and Koller and Friedman [13] are standard texts). In this approach, one represents a joint probability distribution over the set of variables M with a DAG, and the interpretation is that (i) the nodes represent the variables, (ii) the edges represent direct dependencies. Thus, an arrow from x_i to x_j indicates that the value taken by x_j depends on the value taken by x_i. Node x_i is then referred to as a *parent* of x_j, and x_j as a *child* of x_i. The *descendants* set for a node is the set of nodes that can be reached on a directed path from the node, while the *ancestors* set is the set of nodes from which the node can be reached on a directed path. A joint probability distribution p represented as a DAG reflects a conditional independence statement: each variable is independent of its nondescendants in the graph given the state of its parents. In other words,

$$x_i \perp_p Nd_i | Pa_i \;\; \forall i \in M, \tag{A1}$$

where Nd_i and Pa_i are the set of nondescendants and parents of x_i, respectively. Similarly, De_i represents the set of descendants of x_i. A probability distribution of p is *perfect* with respect to a DAG $\mathcal{G}$ if and only if every conditional independence relation under p is also obtainable from $\mathcal{G}$. In this case, p is called DAG-perfect.

Recall that in our modeling framework, p_O is a marginalization of p on the set $O \subset M$, and p is DAG-perfect on M.

Proposition A1 (Chickering et al. [15], Corollary 11)**.** *Given the set M and probability distribution p_O, checking the existence of a Bayesian network DAG for a constant parameter bound is NP-hard.*

It is easy to see that in the notation of Propositions 1 and 2, the set K_α is set of parents, $D_\alpha \backslash K_\alpha$ the set of descendants, and $M \setminus D_\alpha$ is the set of nondescendants of α. That is, $Pa_\alpha = K_\alpha$, $De_\alpha = D_\alpha \backslash K_\alpha$, and $Nd_\alpha = M \backslash D_\alpha$. In the original Chickering et al. [15] setup, the parameter bound refers to the number of conditional probability entries from parent nodes to child nodes. Thus, for example, if a parent node had 2 states, and the corresponding child node had 3 states, this parent–child pair would contribute 6 entries to the parameter count. Because the set of states of the world is finite, this implies a bound on the cardinality of the set of parents in the DAG. Further, since every child has a parent, this implies a bound on the cardinality of the set of descendants in the DAG. Thus, without loss of generality, one can interpret "parameters" in Proposition A1 to mean the set of parents and descendants of the variables. This is the interpretation used in this paper.

In the body of the paper, we have assumed that the joint probability distribution p is representable as a directed acyclic graph. Let me now show that a similar logic goes through even when p might not be representable in this manner. Suppose the probability distribution p was not given to be DAG-perfect. This would be the case, for instance, if the outcomes in the historical database were not optimal. In this case, Chickering et al. [15] proved that checking the existence of a parameter bounded DAG is NP-hard under p.

Proposition A2 (Chickering et al. [15], Theorem 10)**.** *Given the set M and a probability distribution p on M, which may not be DAG-perfect, checking the existence of a Bayesian network DAG for a constant parameter bound is NP-hard.*

Therefore, if we assumed the entire set M to be observable, and substituted p in the place of p_O, we would obtain (following reasoning that closely parallels Theorem 1) that there does not exist a polynomial implementation for the Bayesian update formula under p. Finally, this result goes through even when the observable set is a strict subset of M.

Proposition A3. *Given a probability distribution p over M, if there does not exist a polynomial implementation for the Bayesian update formula under p, there cannot exist a polynomial implementation for the Bayesian update formula under p_O, a marginalization of p over a subset $O \subset M$.*

Proof. The proof is by contradiction. Suppose we did have a polynomial implementation for the Bayesian update formula under p_O. Observe that we are agnostic about the constituent variables of set O as long as it is a strict subset of M. Thus, for any two variables $a, b \in M$, we could always construct a set O that contained a and b and whose cardinality was lower than M's by a constant (say c). In other words, $a, b \in O$, and $|M| - |O| = c$. Given the probability distribution p over M, we could obtain the marginalization p_O over such an O in polynomial time (as it only depends on the number of states of the variables in $M \backslash O$), and then use the polynomial implementation for the Bayesian update formula under p_O. Since p_O is just a marginalization of p, we would therefore have a polynomial implementation for the Bayesian update formula under p. Thus, we arrive at a contradiction. $\square$

References

1. Grossman, S.J.; Stiglitz, J.E. On the Impossibility of Informationally Efficient Markets. *Am. Econ. Rev.* **1980**, *70*, 393–408.
2. Glosten, L.; Milgrom, P. Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *J. Financ. Econ.* **1985**, *14*, 71–100. [CrossRef]
3. Kyle, A. Continuous Auctions and Insider Trading. *Econometrica* **1985**, *5*, 1315–1335. [CrossRef]
4. Athey, S.; Wager, S. Policy learning with observational data. *Econometrica* **2021**, *89*, 133–161. [CrossRef]
5. Chernozhukov, V.; Newey, W.K.; Singh, R. Automatic Debiased Machine Learning of Causal and Structural Effects. *arXiv* **2021**, arXiv:1809.05224.
6. Dao, T.; Kamath, G.M.; Syrgkanis, V.; Mackey, L. Knowledge Distillation As Semiparametric Inference. In Proceedings of the International Conference on Learning Representations (ICLR'21), Vienna, Austria, 28–29 October 2021.
7. Iskhakov, F.; Rust, J.; Schjerning, B. Machine learning and structural econometrics: Contrasts and synergies. *Econom. J.* **2020**, *23*, S81–S124. [CrossRef]
8. Oprescu, M.; Syrgkanis, V.; Wu, Z.S. Orthogonal random forest for causal inference. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 4932–4941.
9. Singh, R.; Sahani, M.; Gretton, A. Kernel Instrumental Variable Regression. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; pp. 4595–4607.
10. Sverdrup, E.; Kanodia, A.; Zhou, Z.; Athey, S.; Wager, S. Policytree: Policylearning via doubly robust empirical welfare maximization over trees. *J. Open Source Softw.* **2020**, *5*, 2232. [CrossRef]
11. Syrgkanis, V.; Lei, V.; Oprescu, M.; Hei, M.; Battocchi, K.; Lewis, G. Machine Learning Estimation of Heterogeneous Treatment Effects with Instruments. In Proceedings of the Conference on Learning Theory, Phoenix, AZ, USA, 25–28 June 2019; pp. 15167–15176.
12. Syrgkanis, V.; Zampetakis, M. Estimation and Inference with Trees and Forests in High Dimensions. In Proceedings of the Annual Workshop on Computational Learning Theory, Graz, Austria, 9–12 July 2020; pp. 3453–3454.
13. Koller, D.; Friedman, N. *Probabilistic Graphical Models*; MIT Press: Cambridge, MA, USA, 2009.
14. Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*; Morgan Kaufman Publishers: San Francisco, CA, USA, 1988.
15. Chickering, D.M.; Heckerman, D.; Meek, C. Large-Sample Learning of Bayesian Networks is NP-Hard. *J. Mach. Learn. Res.* **2004**, *5*, 1287–1330. [CrossRef]
16. Caravagna, G.; Ramazzotti, D. Learning the structure of Bayesian Networks via the Bootstrap. *Neurocomputing* **2021**, *448*, 48–59. [CrossRef]
17. Constantinou, A. Learning Bayesian Networks That Enable Full Propagation of Evidence. *IEEE Access* **2020**, *8*, 124845–124856. [CrossRef]
18. Malone, B.; Kangas, K.; Jarvisalo, M.; Koivisto, M.; Myllymäki, P. Empirical Hardness of Finding Optimal Bayesian Network Structures: Algorithm Selection and Runtime prediction. *Mach. Learn.* **2018**, *107*, 247–283. [CrossRef]
19. Platas-López, A.; Mezura-Montes, E.; Cruz-Ramírez, N.; Guerra-Hernández, A. Discriminative Learning of Bayesian Network Parameters by Differential Evolution. *Appl. Math. Model.* **2021**, *93*, 244–256. [CrossRef]
20. Talvitie, T.; Eggeling, R.; Koivisto, M. Learning Bayesian Networks with Local Structure, Mixed Variables, and Exact Algorithms. *Int. J. Approx. Reason.* **2019**, *115*, 69–95. [CrossRef]
21. Zhang, Y.; Guo, Z.; Rekatsinas, T. A Statistical Perspective on Discovering Functional Dependencies in Noisy Data. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Portland, OR, USA, 14–19 June 2020; pp. 861–876. [CrossRef]
22. Scanagatta, M.; Salmerón, A.; Stella, F. A survey on Bayesian network structure learning from data. *Prog. Artif. Intell.* **2019**, *8*, 425–439. [CrossRef]
23. Rubinstein, A. *Modeling Bounded. Rationality*; The MIT Press: Cambridge, MA, USA, 1998.
24. Spiegler, R. *Bounded Rationality and Industrial Organization*; Oxford University Press: New York, NY, USA, 2011.
25. Ellis, A.; Piccione, M. Correlation Misperception in Choice. *Am. Econ. Rev.* **2017**, *107*, 1264–1292. [CrossRef]
26. Esponda, I.; Puozo, D. Berk—Nash Equilibrium: A Framework for Modeling Agents With Misspecified Models. *Econometrica* **2016**, *84*, 1093–1130. [CrossRef]

27. Eyster, E.; Rabin, M.; Vayanos, D. Financial Markets Where Traders Neglect the Informational Content of Prices. *J. Financ.* **2019**, *74*, 374–399. [CrossRef]
28. Eyster, E.; Piccione, M. An Approach to Asset Pricing under Incomplete and Diverse Perceptions. *Econometrica* **2013**, *81*, 1483–1506. [CrossRef]
29. Jehiel, P. Analogy-based Expectation Equilibrium. *J. Econ. Theory* **2005**, *123*, 81–104. [CrossRef]
30. Jehiel, P. Analogy-Based Expectation Equilibrium and Related Concepts: Theory, Applications, and Beyond. World Congress of the Econometric Society. 2020. Available online: https://philippe-jehiel.enpc.fr/wp-content/uploads/sites/2/2020/10/SurveyABEE.pdf (accessed on 10 May 2021).
31. Jehiel, P.; Koessler, F. Revisiting Games of Incomplete Information with Analogy-based Expectations. *Games Econ. Behav.* **2008**, *62*, 533–557. [CrossRef]
32. Mailath, G.J.; Samuelson, L. Learning under Diverse World Views: Model-Based Inference. *Am. Econ. Rev.* **2020**, *110*, 1464–1501. [CrossRef]
33. Spiegler, R. Bayesian Networks and Boundedly Rational Expectations. *Q. J. Econ.* **2016**, *131*, 1243–1290. [CrossRef]
34. Steiner, J.; Stewart, C. Price distortions under coarse reasoning with frequent trade. *J. Econ. Theory* **2015**, *159*, 574–595. [CrossRef]
35. Williamson, D.P.; Shmoys, D.B. *The Design of Approximation Algorithms*; Cambridge University Press: New York, NY, USA, 2011.
36. Esponda, I. Behavioral Equilibrium in Economies with Adverse Selection. *Am. Econ. Rev.* **2008**, *98*, 1269–1291. [CrossRef]
37. Aumann, R. Interactive Epistemology I: Knowledge. *Int. J. Game Theory* **1999**, *28*, 263–300. [CrossRef]
38. Ralston, A. De Bruijn Sequences—A Model Example of the Interaction of Discrete Mathematics and Computer Science. *Math. Mag.* **1982**, *55*, 131–143. [CrossRef]
39. Piccione, M.; Rubinstein, A. Modeling the Economic Interaction of Agents with Diverse Abilities to Recognize Equilibrium Patterns. *J. Eur. Econ. Assoc.* **2003**, *1*, 212–223. [CrossRef]
40. Milgrom, P.; Stokey, N. Information, trade and common knowledge. *J. Econ. Theory* **1982**, *26*, 17–27. [CrossRef]