Algorithm Games and Rational Play with Strategic Inference

In-Koo Cho* and Jonathan Libgober**

*Emory University
**University of Southern California

September 3, 2021

Abstract. We develop an approach, inspired by ideas from machine learning, to study how algorithms can induce rational play in settings featuring sequential moves and exogenous uncertainty. We consider an algorithm designer seeking to learn and prescribe actions for a second-mover using only historical data. While constraints on feasible algorithms may prevent as-if rational play from emerging, we describe how these constraints can be circumvented to guide behavior across a rich set of possible environments using limited details. Provided a condition known as *weak learnability* holds, Adaptive Boosting algorithms can induce behavior that is (approximately) as-if rational, across a wide range of environments.

1. Introduction

When might "as-if" rational play emerge in strategic settings when rationality is not assumed? This paper addresses this classic question within a class of sequential move games featuring a lemons problem—specifically, when the action of a first mover could convey information about an underlying (payoff-relevant) state to a second mover. The first mover chooses their strategy with commitment as a function of the state (as in Kamenica and Gentzkow (2011), among others). Thus, rationality requires making an inference regarding the sender's strategy (i.e., a *strategic inference*).

Contact. icho30@emory.edu, libgober@usc.edu . This paper was previously circulated under the title "Machine Learning for Strategic Inference." We thank Juan Carrillo, Jason Hartline, Navin Kartik, Roger Moon, Xiaosheng Mu, Guofu Tan, Ashesh Rambachan, Joel Sobel, Erik Strand, and especially Grigory Franguridi, for helpful conversations and comments, and seminar audiences at AMETS, Emory, the Econometric Society World Congress, the NSF/NBER/CEME Conference on Mathematical Economics, Rochester, UC-San Diego, and USC. This project started when the first author was visiting the University of Southern California. We are grateful for hospitality and support from USC. Financial support from the National Science Foundation is gratefully acknowledged.

We consider the design of algorithms with the goal of inducing behavior that approximates rationality across a wide set of parameter specifications, within this class of interactions.

There are two main novelties to our exercise. First, we introduce the structure of an *algorithm game* to study the selection of strategies in this class of sequential move games. The concept of an algorithm game is quite close to the concept of a "machine game" as studied in Rubinstein (1986) and Abreu and Rubinstein (1988), which addresses the above question in the context of two-player repeated games. Those papers studied the construction of automata that could play particular dynamic strategies in repeated games settings. Automata, in turn, are reductive descriptions of strategies, evaluated in terms of the payoffs they induce in the repeated game. In an algorithm game, an algorithm is chosen in order to construct a strategy to be used in the sequential move interaction, with the goal of inducing play which performs well from a payoff perspective. Like automata, algorithms react to what happens in the course of the game; unlike automata, however, algorithms we study produce a strategy which will generally depend on an underlying state distribution (and thus involve learning about exogenous uncertainty, which is absent from the settings of Rubinstein (1986) and Abreu and Rubinstein (1988) and those that follow, to our knowledge). As such, having a longer history will naturally allow better inference, and thus improve the performance of alogirthms—a feature that does not emerge in machine games.

The second novel aspect is that the strategies implemented by algorithms are endogenously constructed by them, and not specified in advance by the modeler. A large literature has studied particular algorithms in simultaneous-move games (e.g., fictitious play, regret-matching etc.), and asked what kind of behavior emerges in the long run. Most of this literature is focused on simultaneous move games without uncertainty,[1] in contrast to our focus on a sequential move game with exogenous uncertainty. More significantly, however, is that we study how algorithms can expand the set of feasible strategies in order to implement play that approximates a rational equilibrium. Algorithms such as fictitious play dictate that some action be chosen within the initial action space of the game. In sender-receiver games, however, the space of receiver strategies is quite large, as it involves specifying an action for every possible message of the sender. In an algorithm game, the algorithm determines the set of feasible strategies. In particular, the specification of the strategy set might only occur during the course of play (and thus be endogenous). Our main result provides a sense in which algorithms can expand the strategy set without increasing algorithm complexity according to a natural measure, in a way we make formally in Section 3.

In this paper, we will further focus on the case where the first-mover chooses rationally.

[1] An important exception is Fudenberg and Kreps (1995b).

While we believe this case is natural for a first step to study sequential move settings with strategic inference—so that we do not need to consider complications associated with interacting algorithms[2]—here it will also allow us to highlight the issue of *algorithm exploitation*. Settings such as ours are naturally analyzed assuming that the second mover best responds to the first, and the first mover optimizes their strategy taking this into account. However, if the first mover is rational and anticipates that the second mover might not always best respond, then the first-mover might seek to induce further departures from Nash behavior in order to exploit such limitations. Furthermore, as there is a fundamental asymmetry between the players in the class of games we study, the question of how to design algorithms for the first mover is ultimately quite different—to maintain focus, we thus take it as given that the first mover achieves rationality immediately.

Our main question is whether the algorithm game possess an equilibrium where the algorithm designer successfully induce a best reply for the second mover (in the long run), and where the first mover chooses a strategy as if the second mover were rational (and therefore, does not seek to exploit the second mover, as mentioned above). If this is possible, then the natural question is how to specify algorithms which achieve this end, and further study what other attractive properties they may possess. Before turning to our results, we mention two other features which elucidate our contribution.

First, while not assumed by the structure of an algorithm game, we specify algorithms concretely in terms of the set of "optimal replies" that the algorithm has the ability to determine, given any observed history. We do this in order to precisely describe the *computational* complexity of algorithms, and to discuss how constraints and capabilities of algorithms influence the ability to approximate rationality. To be more precise, the algorithms we evaluate are those that are initially endowed with a set of what we refer to as *baseline strategies*, and the ability to find the "best performing strategy" (according to an arbitrary metric, which could be depend on the history of the interaction) from this set. We posit that an algorithm that involves a larger set of baseline strategies is more complex, as finding the best performing one becomes more computationally demanding if there are many possible candidates.

We believe this focus is helpful toward our goal of extending the literature on algorithmic implementations of rational behavior to the settings we study here; indeed, as alluded to above, this exercise necessitate the algorithm producing a response for *every* possible first-mover action. This lends itself sensible restrictions on which strategies players might use (e.g., monotonicity

[2] Several authors have noted that outcomes may depart from Nash predictions when multiple algorithms interact; see, for instance, Calvano, Calzolari, Denicolò, and Pastorello (2019) or Brown and MacKay (2021), who focus in particular on competitive pricing algorithms. The algorithms in these papers allow strategies to condition on past actions of the other players, a feature which drives many of their observations, most notably the possibility of "collusive" (non-Nash) outcomes. Our setting similarly requires the algorithm to "reply" to the first mover, and thus condition on the past actions that were observed. This feature is absent from the past work mentioned above.

requirements), and also increases the importance of understanding how those restrictions influence algorithm performance.

Constraints on algorithms of the kind in our model are often studied in the machine learning literature, which typically treats the data generating process as exogenous. Our goal, however, is to perform a similar algorithm design exercise, but where the "data" is generated by endogenous strategic choices. The endogeneity will influence the realized, on-path performance of algorithms, even though we will need to consider off-path performance as well to determine optimal choices. That said, to make sense of the computational constraints on algorithms, it may be instructive to note that in typical machine learning problems, a simple prediction (for instance, a "yes-no" recommendation) is sought for an observation among a very large set of possibilities. Seeking to find the correct recommendation for each one may be intractable or undesireable (given data limitations), and so a simpler set may be used as a baseline. Still, an analyst might still be able to construct a new classification rule by specifying how this should be done in advance. For us, this takes the form of assuming the algorithm is limited in what can be fit to the data, but has other capabilities, as we will discuss.

On this note, the second notable feature of our approach is that we draw inspiration from the machine learning literature to define "good performance" of an algorithm. The requirement we use is that the algorithm can achieve a *Probably Approximately Correct (or PAC)* guarantee. While this notion is familiar from some of the literature surveyed below, to our knowledge it has not been studied in the particular literature our exercise falls under. To understand this notion, consider two reasons why an algorithm may produce a recommendation which differs from rationality. First, it may be that the algorithm hasn't yet seen enough data on the first-movers action to precisely provide a correct recommendation. Second, it may be that the algorithm hasn't figured out how to reply to some "unlucky" actions, even if it has figured out the correct response following most actions. The PAC requirement states that both of these are low probability events, provided a reasonable[3] amount of data.

As said above, our interest in this paper is in determining whether the algorithm game has an equilibrium whereby the algorithm induces (an appropriate notion of) rationality. Our analysis seeks to highlight the following tension. On the one hand, given the problem of algorithm exploitation mentioned above, it is generally not possible to achieve this goal if the set of strategies the algorithm can prescribe coincides with the set of baseline strategies.[4] Nevertheless, it is possible to induce rationality if algorithms can endogenously expand the set of strategies they can implement, even without enriching the set of baseline strategies. In this case, the challenge is to

[3] More formally, an amount of data that is polynomial in the parameters.

[4] Recall that the set of baseline strategies is the set from which the algorithm has the capability of finding a "best-fitting strategy," given the historical data

specify how this expansion should be done, without assuming richer abilities to determine optimal strategies. This shows that the second main novelty of our exercise is crucial for delivering our results.

We describe some details regarding the kind of algorithm exploitation we focus on in this paper. The key economic force, related to the restrictions on baseline strategies, was first identified (to our knowledge) by Rubinstein (1993). That paper studied a buyer-seller game where the buyer's possible strategies are restricted. Specifically, this paper showed that if a *rational* decisionmaker is restricted to use a single threshold classifier—i.e., one that makes the same decision on a given side of a fixed threshold—then the seller can price discriminate via a particular form of randomization which "fools" these buyers into making a decision which is suboptimal given the realized price.[5]

Our framework nests Rubinstein (1993) as a special case, and we use this as a running example to highlight our main results and why we believe they are of interest. However, our framework accommodates more general environments as well (e.g., Bayesian Persuasion). Rubinstein's paper, however, assumes that (1) the set of strategies a buyer can use is limited, but (2) given the seller's strategy, the optimal decision rule is automatically chosen. By contrast, our exercise allows (1) algorithms to *construct* more elaborate strategies, despite only being able to *find* an optimal rule within a limited set, and (2) the optimal rule to be determined using historical data, and not by fiat. We call a *baseline strategy* one that belongs to the limited set the algorithm can optimize among (so that in (Rubinstein 1993), the baseline strategies are the aforementioend single-threshold classifiers).

Our proposal is a version of the Adaptive Boosting algorithm (Schapire and Freund (2012)), which specifies how to define a strategy as a "weighted combination of baseline strategies," with the weights specified by the algorithm. The algorithm requires the ability to (repeatedly) find an optimal response to an arbitrary conjecture of the first-mover's play. The requirement on the set of baseline strategies for this algorithm to work is known as *weak learnability.* This requirement states that it is only necessary for the optimal baseline strategy to uniformly outperform a random guess. This requirement is significantly less demanding than requiring that the strategy always produce the optimal reply (which rationality requires). We provide results which show how to check weak learnability straightforwardly in applications, particularly when resorting to single-threshold classifiers (which typically have natural interpretations, even beyond Rubinstein (1993)).

[5] The reasoning behind this result is as follows. First, the optimally chosen single-threshold classifier can do strictly better than simply randomizing the guess, implying that the seller can exploit the incentives of the buyer in order to manipulate the decision rule. On the other hand, it is impossible for threshold rules to implement the optimal decision with probability 1 when this rational rule is non-monotone in the price. The first point implies the buyer trades off against errors, and the second point implies that the tradeoff falls short of the fully rational response. As a result, the seller can force a different decision than would be rationally optimal for these buyers (with arbitrarily high probability).

To summarize, then, the answer to our theoretical question is that rationality can be ensured with the ability to (a) find an optimal strategy among a set of strategies satisfying weak learnability, and (b) combine strategies in a particular (pre-specified) way. At first glance, one might conjecture a significant gap between a set of baseline strategies satisfying weak learnability versus those that can induce rationality. Rationality requires very rich strategies to be used, and for their performance to leave very little room for error. Weak learnability only requires a uniform improvement over a random guess. It is therefore perhaps surprising that in our exercise, there is no gap at all.

It is worth emphasizing one technical difference—due to our focus on a strategic inference problem—between our exercise and similar ones considered in computer science or machine learning, where these issues have received attention. In principle, the rational decision in our model is *not* observed if the first-mover uses a strategy that does not reveal the state given an observation. In a buyer-selling setting, for instance, it may be that "low quality" is observed at some price, but that "high quality" is in fact more likely and that correspondingly a rational buyer (receiver) would choose a "buying" action. In our problem, the payoff-maximizing decision must be *inferred* and constructed by the algorithm. One of the main results of this paper is that this added difficulty does not change the qualitative desirable properties of the algorithm. Showing this uses results from large deviations theory, which we use to describe the modifications on the exact rate at which we can guarantee an approximation of the rational strategy.

1.1. Literature

This paper takes the framework of PAC learnability, familiar from machine learning, and applies it to a strategic setting. Within economics, this agenda is most closely related to the literature on learning in games when behavior depends on a statistical method. The single-agent problem is a particular special case, and this case is the focus of Al-Najjar (2009) and Al-Najjar and Pai (2014). However, since we are focused on a strategic setting, the data the algorithm receives is *endogenous* in our setting. In contrast, their benchmarks correspond to the case of exogenous data. This problem is also studied in Spiegler (2016), who focuses on causality and defines a solution concept for behavior that arises from individuals fitting a directed acyclic graph to past observations. Eliaz and Spiegler (Forthcoming) study the problem of a statistician estimating a model in order to help an agent take an action, motivated as we are by issues involved with the interaction between rational players and statistical algorithms. More recently, Zhao, Ke, Wang, and Hsieh (2020) take a decision-theoretic approach in a single-agent setting with lotteries, showing how a relaxation of the independence axiom leads to a neural-network representation of preferences.

Classic contributions to the literature on algorithm approximations of Nash behavior include Hart and Mas-Colell (2000), Foster and Vohra (1997) and Fudenberg and Levine (1995); see also

Fudenberg and Levine (1998) for a textbook treatment of this area. Yet even recently, the literature has still for the most part focused on settings where the interactions between players is *static*, ruling out the main environments we are interested in here. In contrast, our setting is a simple, two-player (and two-move) sequential game. Cherry and Salant (2019) discuss a procedure whereby players' behavior arises from a statistical rule estimated by sampling past actions. This leads to an endogeneity issue similar to the one present in our environment, i.e., an interaction between the data generating process and the statistical method used to evaluate it. Liang (2018) also focuses on games of incomplete information, asking when a class of learning rules leads to rationalizable behavior. Studying model selection in econometrics, Olea, Ortoleva, Pai, and Prat (2019) consider an auction model and ask which statistical models achieve the highest confidence in results as a function of a particular dataset.[6] By contrast, we show that a version of the weak learnability condition in settings with two possible receiver actions also applies to settings with an arbitrary finite number of actions. Our ability to handle this in our problem suggests our proposals are of broader interest. We believe that this extension is important, as it shows our conclusions do not hinge on other artificial limitations on the environment.

The literature on learning in *extensive* form games has typically assumed that agents experiment optimally, and hence embeds notion of rationality on the part of agents which we dispense with in this paper. Classic contributions include Fudenberg and Kreps (1995a), Fudenberg and Levine (1993) and Fudenberg and Levine (2006). Most of this literature has focused on cases where there is no exogenous uncertainty regarding a player's type, and asking whether self-confirming behavior emerges as the outcome. An important exception is Fudenberg and He (2018), who study the steady-state outcomes from experimentation in a signalling game (see also Fudenberg and Clark (2021)). While a rational agent in our game would need to form an expectation over an exogenous random variable, issues related to off-path beliefs do not arise because our sender has commitment.

Perhaps closest in motivation is the computer science literature studying algorithm perform in strategic situations. Braverman, Mao, Schneider, and Weinberg (2018) consider optimal pricing of a seller repeatedly selling to a single buyer who repeatedly uses a no-regret learning algorithm. They show that, on the one hand, while a particular class of learning algorithms (i.e., those that are *mean-based*) are susceptible to exploitation, others would lead to the seller's optimal strategy simply being to use the Myersonian optimum. Deng, Schneider, and Sivan (2019) also study strategies against no-regret learners in a broad class of games without uncertainty, and consider whether a strategic player can guarantee a higher payoff than what would be implied by first-mover advantage. Blum, Hajiaghayi, Ligett, and Roth (2008) consider the Price of Anarchy (i.e.,

[6] On the question of algorithms in particular, one concern is that the algorithm design problem may be susceptible to bias or induce unwanted discrimination when implemented, relative to rationality. See Rambachan, Kleinberg, Mullainathan, and Ludwig (2020) for an analysis of these issues and how they may be overcome.

the ratio between first-best welfare and worst-case equilibrium welfare), and show in a broad class of games that this quantity is the same whether players use Nash strategies or regret-minimizing ones. Nekipelov, Syrgkanis, and Tardos (2015) assume players in a repeated auction use a no-regret learning algorithm, making similar behavioral assumptions as we do here. Their interest is in inferring the set of rationalizable actions from data.

Though we seek to incorporate several aspects of this literature's conceptual framework, our problem has three notable differences. First, this literature typically assumes particular algorithms or principal objectives (such as no-regret learning) which differ from traditional Bayesian rationality. In contrast, we maintain a Bayesian rational objective for the seller, and also focus on an algorithm designer seeking to maximize expected payoffs. Second, we focus on relating the incentives of the rational player and *the algorithm's capabilities*, and study the extent to which different assumptions on the algorithm design problem influence the task of approximating rationality. Our main result articulates how different action spaces for the algorithm designer yield different results regarding whether and when the outcome will approximate the rational benchmark. Lastly, our general framework focuses on settings with strategic inference—that is, where the payoffs following a given first-mover action are state-dependent—and thus covers a set of single-agent applications which extend beyond particular pricing settings, where most (though admittedly not all) of this literature has focused. In particular, the settings discussed in this literature typically do not cover lemons markets settings, to the best of our knowledge.[7] As a result, new technical issues (e.g., dealing with residual uncertainty in the correct actions) is not addressed in these papers. Despite these differences, our hope is that this paper inspires further connection between the economics literature on decisionmakers as statisticians and the computer science literature on strategic choices against classes of algorithms. It appears to us that these results from computer science have not yet been fully appreciated in economics.

[7] An important exception is Camara, Hartline, and Johnsen (2020), who study an environment covering many of our same applications such as Bayesian Persuasion. However, they still maintain the other two distinguishing features, focusing on a regret objective for the principal, as well as particular no-regret assumptions for the agent. Still, we emphasize that both our paper and theirs focuses on environments where the principal/sender chooses a state-dependent strategy. This leads to the aforementioned endogeneity between the data generating process (induced by the principal) and the choices of the algorithm/learner—this emerges due to the fact that the same sender action may induce two distinct replies from the algorithm following two distinct sender strategies. In their setting, this endogeneity motivates the use of "policy-regret" as an objective for the principal (due to their reinforcement learning approach to the principal's problem). While we do not use a regret objective for the principal, see Arora, Dekel, and Tewari (2012) and Arora, Dinitz, Marinov, and Mohri (2018) for more on the differences between these notions.

2. Environment

Our model consists of two components; a stage game and a supergame. Strategies are chosen in the latter and then executed in the former. We call our particular supergame an *algorithm game*, as this is where the choice of algorithm is made. We defer a discussion of how we represent algorithms, as well as the desireable features of our proposed algorithm, until Section 3.

2.1. Stage Games

2.1.1. Actions and Parameters

The stage game is a sender-receiver game in which an informed sender makes the first move. We often call the sender *the (informed) principal*, and the receiver *the agent*, as in Maskin and Tirole (1992), though our model also describes a sender-receiver game with sender commitment, as in Kamenica and Gentzkow (2011).

Let Θ denote a set of types endowed with a prior distribution π, where $\pi(\theta)$ is the probability that type $\theta \in \Theta$ is realized. This type is payoff relevant to both the sender and the receiver. Throughout the paper, we only consider π with finite support. We also assume throughout that π is known by the sender; we take π to be (commonly) known by the receiver only in the benchmark where he is (assumed to be) rational, though most of the paper is not concerned with this case.

Conditioned on the realized value of $\theta \in \Theta$, the sender takes an action $p \in \mathcal{P} \subset \mathbb{R}^n$ where $\mathcal{P}$ is compact. Our main analysis will assume $|\mathcal{P}| < \infty$, but we discuss how to modify this assumption in Section 6.3.2. The strategy of the Sender is:

$$\sigma : \Theta \to \Delta(\mathcal{P}),$$

where $\Delta(X)$ denotes the set of probability distributions over a set X. We let Σ denote the set of all possible σ. The choice of $\sigma \in \Sigma$ is determined in the Algorithm Game described below in Section 2.2.1. At the start of the stage game, p is drawn (as per σ) and observed by the receiver. Conditioned on p (but not θ), the receiver chooses $a \in A$ according to a strategy:

$$r : \mathcal{P} \to \Delta(A).$$

We assume $|A| < \infty$, and in our analysis we treat the case of $|A| = 2$ and $|A| > 2$ separately. Stage game payoffs of Sender and Receiver following (θ, p, a) are $u(\theta, p, a)$ and $v(\theta, p, a)$, respectively.

The timing of the moves in the stage game is as follows:

S_1. The state $\theta \in \Theta$ is realized according to π, with only Sender observing θ.

S_2. Sender's action $p \in \mathcal{P}$ is realized according to $\sigma(p:\theta)$.

S_3. The receiver takes action $a \in A$ conditioned on p (but not θ).

S_4. Payoffs are realized according to $u(\theta, p, a)$ and $v(\theta, p, a)$.

Though special, this stage game framework is very rich and covers many different previously studied applications. For instance, If we interpret $p = (p_1, \ldots, p_n)$ as a contract, and $a \in A = \{-1, 1\}$ as "reject" ($a = -1$) or "accept" ($a = 1$), the stage game is a model of the informed principal (Maskin and Tirole (1992)). If p is interpreted as a message sent by a worker, and $a \in A$ as the wage paid by the firm, then the stage game becomes a signaling game (Spence (1973)). Both of these are discussed in more detail in Section 5. For now, we place no further restrictions on $u(\theta, p, a)$ and $v(\theta, p, a)$, though these are often implicit in the economic problem of interest.

2.1.2. Payoffs and the Rational Benchmark

The outcomes of the above interactions when both players are rational is familiar. In that case, Receiver's optimization problem is:

$$\max_{a \in A} \sum_{\theta \in \mathsf{supp}\pi} v(\theta, p, a)\pi(\theta : p)$$

where $\pi(\theta : p)$ is the posterior probability assigned to θ conditioned on p. If p is used with a positive probability by σ, then $\pi(\theta : p)$ is computed by Bayes rule:

$$\pi(\theta : p) = \frac{\sigma(p : \theta)\pi(\theta)}{\sum_{\theta'} \sigma(p : \theta')\pi(\theta')}.$$

We will refer to the *rational label*, denoted $y^R : \Sigma \times \mathcal{P} \to A$, as the solution to the following optimization problem:[8]

$$\sum_{\theta \in \mathsf{supp}\pi} v(\theta, p, y^R(\sigma, p))\pi(\theta : p) \geq \sum_{\theta \in \mathsf{supp}\pi} v(\theta, p, a)\pi(\theta : p) \qquad \forall a \in A,$$

where $\pi(\theta : p)$ is computed via Bayes rule whenever $\sum_\theta \sigma(p : \theta)\pi(\theta) > 0$.[9] Let $\mathcal{H}^R$ denote the set of all y^R which emerge under some $\pi \in \Pi$.

[8] We will occasionally use the term "label" to distinguish the strategy of the receiver, as both sender and receiver will use strategies. We also view this as helpful in relating our work to the Machine Learning literature, where we derive a lot of the motivation for our exercise, as it is consistent with the usage there.

[9] For a fixed σ, $y^R(\sigma, \cdot) : \mathcal{P} \to A$ is a strategy of the agent, satisfying sequential rationality.

We will let σ^R denote the best response of the sender against a Bayesian rational receiver with perfect foresight:

$$\sum_{\theta,p,a} u(\theta,p,a)\sigma^R(p:\theta)y^R(\sigma^R,p)\pi(\theta) \geq \sum_{\theta,p,a} u(\theta,p,a)\sigma(p:\theta)y^R(\sigma,p)\pi(\theta) \qquad \forall\sigma\in\Sigma.$$

Note that (σ^R, y^R) constitutes a perfect Bayesian equilibrium in the stage-game with a rational receiver. Of course, if the receiver were not rational, then it correspondingly might not be optimal for a rational sender to choose σ^R.

2.1.3. Running Example: Monopoly Market (Rubinstein 1993)

As discussed in the introduction, one of the most similar exercises to ours was performed by Rubinstein (1993). That paper considers a particular buyer-seller setting, which we now describe, but assumes that the buyer's strategy is chosen optimally from a restricted set of strategies. Aside from the added generality of our model, the main differences between that paper and ours is the imposition of the structure of an algorithm game. We comment further on the algorithm game after it is introduced.

Rubinstein (1993) considers a setting where a monopolist must choose a single price to sell in two different markets—or equivalently, two different kinds of buyers (only one of which is guided by the algorithm, the other of which is rational). The algorithm, by contrast, decides whether the buyers should buy (i.e., choose $a = 1$) or not buy (i.e., choose $a = -1$), given the observed price. The quality of the seller's product is given by $\theta \in \{L, H\}$, and the buyer's value for the product is v_θ, where $v_H > v_L$. This is a standard lemons setting, since the price (given the seller's strategy) might reveal information about the product quality. Notice that:

$$y^R(\sigma,p) = 1 \Leftrightarrow \mathbb{E}[v_\theta \mid p,\sigma] \geq p.$$

Importantly, since the seller is constrained to use the same price in both markets, their payoff is not simply 0 if the buyer chooses not to buy. In particular, Rubinstein (1993) considers a seller[10] profit function with three additional properties:

- $u(L,p,-1) < u(L,p,1)$
- $u(H,p,1) < u(H,p-1)$
- $\arg\max_p u(L,p,1) = v_L < \arg\max_p u(H,p,-1) \leq v_H$

[10] Or, in our terminology, sender.

The first condition says that when the product quality is low, the seller would rather the buyer buy (e.g., if production costs are 0). The second says that the seller would rather have the buyers *not* buy when the product quality is high (e.g., if the buyers serviced by the algorithm are costly to serve, at least on average).

The last condition describes how the seller would like to price in this setting, and what they would try to get the algorithm to do—ideally, they would like to have (1) the buyer purchase when $\theta = L$, charging a low price, and (2) the buyer *not* purchase when $\theta = H$, while choosing a high price. Importantly, such a policy is not implementable as a unique equilibrium with rational buyers—in that case, the high price would reveal high quality.

To conclude, we note that one can show that σ^R randomizes over at most two prices. $\mathcal{H}^R$ consists of the set of all *increasing single-threshold strategies*:

$$h_{\overline{v},k} = \begin{cases} 1 & p \geq \overline{v} \\ -1 & p < \overline{v} \end{cases}, \text{ where } k \in \{-1, 1\}.$$

The set of single-threshold strategies (which may either be increasing or decreasing) will play a role in our main results below. To appreciate them, however, for now we simply note that this set contains all rational replies the buyer may use, to any seller strategy.

2.2. Algorithm Game (i.e., the Supergame)

Our interest is in whether rational replies can emerge as a long run outcome in repeated play of the stage game. We dub the corresponding supergame as an *algorithm game*. Specifically, we consider a repetition of the stage game interaction, played over discrete time $t = 1, 2, \ldots$, where the stage game interactions occur at every $t \geq 1$. At each time t, we will take the true state θ to be drawn IID across periods according to π; in addition, we let (p_t, a_t) denote the actions of the sender and receiver, respectively.

2.2.1. Defining Algorithms

Throughout this paper, we assume that the sender (principal) is fully rational, but the strategic choice of the receiver (agent) must be delegated to an algorithm. The algorithm will take in the set of *histories*, a sequence of outcomes that occurred during the game. We denote the set of histories by $\mathcal{D}$, and denote the particular history of outcomes until time t by D_t. Our two main results, Propositions 5 and 6, will correspond to each of the following two cases:

- $D_t = \{(p_1, y^R(p_1, \sigma)), \ldots, (p_t, y^R(p_t, \sigma))\}$ (Proposition 5)

- $D_t = \{(p_1, \theta_1), \dots, (p_t, \theta_t)\}$ (Proposition 6)

In the first case, the rational action is itself observed, while in the second case the latent state is observed. Our view is that the second assumption is less strong than it may initially appear; one might think a more realistic assumption would be that a decisionmaker observes their own payoffs, possibly without revealing the whole state θ itself. While in some cases it may be reasonable to assume the state is observed ex-post, we seek to to shut down experimentation motives for the algorithm designer, which richer model setups would be able to accommodate more easily. To see what we mean by this, note that in principle, the algorithm could recommend an arbitrary action $a \in A$, and thereby learn $v(\theta, p, a)$ from the ability to see the receivers payoffs. Collapsing all $\theta \in \Theta$ which give identical values of $v(\theta, p, a)$ for all a, this would allow the algorithm to observe θ itself. So, our assumption that θ_t is observable is meant to allow us to focus on the case where the rational reply must be *inferred* ont he one hand, but on the other hand the algorithm does not need to consider experimentation motives of the algorithm.[11] Which case is more natural is problem specific; however, notice that in the second case, the rational reply needs to be inferred, making this case the "harder" one.

Definition 1. *Let Γ be some fixed set of possible (receiver) strategies. A statistical procedure or algorithm is a function*

$$\tau : \mathcal{D} \to \Gamma.$$

where $\mathcal{D}$ is a set of histories.

We let $\mathcal{T}$ denote the set of feasible algorithms (i.e., a subset of the set of functions from $\mathcal{D}$ into Γ); in other words, the choice set of the algorithm designer is $\mathcal{T}$. Feasibility in this case reflects what the algorithm designer is capable of selecting. In many cases, $\mathcal{T}$ may be retricted, or only implicitly defined[12]; however, in the general structure of an algorithm game, no particular restrictions are assumed.

Our main interest in this paper is in understanding which kinds of Γ and $\mathcal{T}$ enable the receiver to approximate the rational label, y^R. Restrictions on $\mathcal{T}$ will come in the form of computational constraints on algorithms, which we discuss more in Section 3.2.

[11] For instance, if we instead had imagined the algorithm provided recommendations on behalf of a large number of receivers simultaneously, then it could give some receivers a "fake" recommendation to ensure all actions were explored. For simplicity, we do not try to extend the model to study this formally, instead simply assuming that ex-post payoffs are observed following each action.

[12] For instance, below we define *recursive ensemble algorithms*, which is one such implicit restriction $\mathcal{T}$

2.2.2. Timing and Objectives

An algorithm game is a *simultaneous move* game between the (rational) sender and an *algorithm designer*, in which the strategies dictating play at times $t = 1, 2, \ldots$ are determined:

A_{-1}. According to some prior distribution, Nature selects the distribution π of the underlying game from a set Π, where Π is some subset of $\Delta(\Theta)$ that only includes distributions with finite support.[13]

A_0. Conditioned on realized π, the sender commits to a strategy $\sigma \in \Sigma$. The receiver (or alternatively, an entity acting on the receiver's behalf) commits to an algorithm $\tau \in \mathcal{T}$ without observing the realized $\pi \in \Pi$. These are chosen simultaneously.

A_1. The stage game is played at time $t = 1, 2, \ldots$, with the receiver's strategy in each period t being $\tau(D_t)$, and therefore his action being $\tau(D_t)(p)$, with the algorithm adding the observation (which includes p and ex-post utility following each receiver action) to the dataset at the end of each period.

We conclude by specifying payoffs in the algorithm game. Given a sequence (θ_t, p_t, a_t), the rational sender's payoff is simply the long run average expected payoff:

$$\mathcal{U}_s(\sigma, \tau) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{E} u(\theta_t, p_t, a_t),$$

where (p_t, a_t) is generated by (σ, τ) in period t and the expectation is otherwise conditioned only on $\pi \in \Pi$ (recalling that θ is taken to be drawn IID). The objective of the rational sender is to maximize $\mathcal{U}_s$ by choosing σ, conditioned on $\pi \in \Pi$. The ex-post payoff of the algorithm designer is also the long-run average expected payoff of the buyers:

$$\mathcal{U}_r(\sigma, \tau) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{E} v(\theta_t, p_t, a_t).$$

[13] Notice that we do not necessarily assume that any pair $\pi_1, \pi_2 \in \Pi$ have intersecting or even overlapping support (though this is also certainly allowed). Correspondingly, we emphasize we do not assume Θ is itself finite, even though all $\pi \in \Pi$ have finite support. If $|\Theta| = \infty$, clearly Π will need to be a strict subset of $\Delta(\Theta)$; if $|\Theta| < \infty$, Π may or may not be a strict subset of $\Delta(\Theta)$.

2.2.3. Revisiting the Running Example and the Possibility of Algorithm Exploitation

Before we turn to a more complete description of how we describe algorithms and $\mathcal{T}$, we continue our running example to illustrate the nontriviality of our exercise and clarify some modelling choices. A natural conjecture about how to specify Γ, as simply as possible, would be the following:

- Find a simple class of strategies which includes every possible equilibrium buyer strategy, and
- Have an algorithm pick the strategy within this class that minimizes the probability of making a mistake.

If the seller's strategy were exogenously given by σ^R, such a scheme would work. In this case, recall that $\mathcal{H}^R$ is the set of single threshold strategies. However, with endogenous σ^R, such a scheme is susceptible to exploitation:

Proposition 1. *Suppose that $\Gamma = \mathcal{H}^R$. There exists a specification of the monopoly model such that no algorithm can construct an optimal reply to the seller's optimal on-path strategy.*

The Appendix describes the conditions necessary, but the intuition is simple.[14] Toward contradiction, if the conclusion were false, then the buyer's optimal strategy would be of a single-threshold form. Recall that the seller's ideal is to charge a high price, but for the algorithm to recommend the buyer not purchase. However, the seller is also able to randomize in such a way that the $y^R(\sigma, p)$ is *non-monotone*. As a result, a buyer constrianed to use a single-threshold would necessarily make an error (since any single-threshold rule is monotone). It turns out that the seller can make sure that, if the buyer were *optimally* trading off errors, they would choose -1 following a high price, as they would wish. By carefully choosing a strategy, this can be done while minimizing other losses, thereby being a profitable deviation.

The idea, then, is that the seller uses a strategy that requires a slightly richer strategy space for the buyer's rational reply, and doing so exploit the buyer's limitations. As per the above, the rational reply to the seller's strategy requires two thresholds, even though the buyer can only use one. But this is actually immaterial, since if the buyer could use two thresholds, then the seller might use a strategy where the rational reply requires three (and so on and so forth). We thus obtain the following strengthening of this result:

[14] The key ideas are present in (Rubinstein 1993), though the results in the Appendix are derived in some added generality than his are stated.

Proposition 2. *Suppose the buyer is restricted to using an n-threshold strategy, for n arbitrary but finite.*[15] *Then there exists a specification of the monopoly model such that no algorithm can construct an optimal reply to the seller's optimal on-path strategy.*

The proof requires us to introduce one other environment, besides the one used in the proof of Proposition 1. The original environment delivers the result when the number of thresholds the buyer can use is odd; when this is even, then a constraint emerges in that if all thresholds are used, then the decision must be the same at both ends of the buyer's strategy space. The details of the environment where exploitation can occur given an even number of thresholds is spelled out in the Appendix.

We also record that, given some specifications, it may indeed be possible to design an algorithm which *outperforms* rationality:

Proposition 3. *There exists a specification of the monopoly model (i.e., a choice of u, v and Θ) such that the second mover can outperform $y^R(\sigma^R, p)$ by using $\Gamma \subset \mathcal{H}^R$.*

The specifications which deliver this are ones where the algorithm only needs to find an optimal reply in a single environment. In these cases, the algorithm can outperform rationality by committing to a particular decision rule which would not itself be a best reply. However, these proposals would be susceptible to details of the enviornment; provided a sufficiently rich set of strategies might potentially be necessary, such restrictions may very well backfire for some particular $\pi \in \Pi$.

2.3. Discussion of Model Assumptions

Before providing more details of how we represent algorithms and describe our main results, we now discuss a number of our key assumptions made above, and descibe the role they play in the analysis.

Sender Rationality Above, we assume that the stage game's first-mover (sender) is rational, whereas the last-mover (receiver) is algorithmic. This assumption is useful because it allows us to ignore questions related to convergence of the *sender's* strategy to equilibrium, and eliminates any corresponding noise or time dependence in the data generating process used by the (receiver's) algorithm. Also note that, for the question of inducing sender rationality to be interesting, one should likely study a different move protocol—since, for us, the receiver has no knowledge which the sender would seek to adapt to, since the sender observes π (and the receiver does not).

[15] An n-threshold strategy is defined by n numbers, say $v_1, \ldots, v_n$, such that the buyer uses the same action at all $p \in (v_i, v_{i+1})$, taking $v_0 = -\infty$ and $v_{n+1} = \infty$

Intertemporal Preferences We assume that the sender is long-lived, while the algorithm acts on behalf of a sequence of short-lived senders, with both maximizing undiscounted payoffs. Prior versions of this paper considered the case where future payoffs were discounted at rate $\delta < 1$; the main lessons remain valid for δ sufficiently large, although there are some added technical difficulties in the analysis of Section 6.3.2 this introduces.

Notice that if the sender were instead short lived, then she would not necessarily benefit from exploiting the algorithm as in Sections 2.1.3 and 2.2.3.[16] Notice that we could have instead considered the case where there is also one long-lived receiver, but which an algorithm makes choices on behalf of. For instance, an insurance company may seek to automate the approval/rejection of contract offers, an application similar to what we discuss in Section 5.3.1. Whether there is a single receiver or a sequence of receivers is expositional, and which makes more sense will depend on the application.

Observed Data We assume that the data at the end of each period is perfectly observed by the receiver. For instance, Etsy or Amazon might be able to ask consumers exactly how much they liked a recommended product. Given sufficient consumer sophistication regarding seller strategies, it may be possible for the platform to determine y^R exactly. Otherwise, the necessary assumption is that consumers give accurate reports. The assumption that the payoffs following *all* actions are observed is made primarily to avoid the need to consider experimentation (so that Etsy would not need to suggest possibly suboptimal products, simply to find out what the value would be). Our own preference is to view the case of observed y^R as an intermediate step, though we do not doubt there are cases where this assumption may be reasonable.

For buyer-seller interactions, the assumption that data is observed by the algorithm is appropriate if consumers always reported to the algorithm after their purchase. But as mentioned above, our model also applies to cases where there is a single long-lived receiver, in which case the assumption would simply be that the algorithm observes own-payoffs ex-post (as with the problem of granting approval, for instance, in an insurance or loan settings[17]). We remark that noise in the correct action ex-post is known to severely inhibit the performance of our proposed algorithm. As we discuss below, this issue actually still arises here when $u(\theta, p, a)$ is observed instead of y^R; however, we can handle this by providing exponential bounds on the amount of

[16] The added difficulty with short-lived senders is actually that, as discussed above, the short-term best reply would be different depending on how close the algorithm has approximated rationality—although, for the short-lived senders to adapt to the algorithm choice, they would need to somehow know how much (and what) data the algorithm received, which would add further moving parts.

[17] An active policy debate relates to the use of algorithms in determining whether defendants should be held before trial, without resorting to cash bail. Though the assumption of a long-lived sender seems less plausible for this application, it does not seem to be a stretch to adapt use framework to speak to this application.

noise as the amount of data grows large. Richer noise in observations of $u(\theta, a, p)$ is not something our solution can address (though other proposals might work).

Once-and-for-all Choices (and IID data) Our setting also assumes that the sender makes a once-and-for-all choice of strategy σ. While the optimal strategy choice may be endogenous to the algorithm—in that the sender would choose different σ if they anticipated a different τ—once σ is chosen, no further endogeneity issue emerges.

For our purposes, this assumption provides the cleanest comparison with the benchmark where the stage game is played once with rational players (where the stage game is as in Kamenica and Gentzkow (2011), possibly allowing sender messages to influence receiver utility). Our main message is that we can replicate this benchmark when the receiver is algorithmic. How senders can achieve commitment in Bayesian Persuasion is a well-known theoretical issue, which would take us too far afield to address (see, for instance, Best and Quigly (2020)). If the sender could adjust their strategy over time, it is less clear what the appropriate benchmark comparison would be. This assumption also dramatically simplifies our analysis by ensuring that the algorithm's data is IID. The notion of PAC-learnability (described below) is usually defined assuming this, and so a more complex notion would be required more generally. We suspect that this assumption be relaxed somewhat, but the main difficulty is in finding settings where this can be done while maintaining tractability and a sensible benchmark comparison.

3. Algorithm Desiderata

The substance of our exercise is to reconcile two *competing* objectives:

- The algorithm should approach the rational benchmark sufficiently quickly, and
- The algorithm is not too computationally demanding.

The tension between these objectives is clarified by our running example (in Sections 2.1.3 and 2.2.3). A computationally simple algorithm might restrict $\Gamma = \mathcal{H}^R$, the set of rational best replies; but as these sections showed, this should not be expected to approximate rationality.

We now formally describe each of these targets, and describe how these desiderata relate to one another. In doing so, we provide more details of how we specify algorithms τ and the set of algorithms $\mathcal{T}$.

3.1. PAC-Learnability

The first desideratum of an algorithm relates to the amount of data necessary for the algorithm to achieve a close approximation.[18] To motivate this, consider the case where the algorithm does, in fact, provide an incorrect recommendation. One could attribute this to two different kinds of failings:

- The particular history realized did not enable the algorithm to approximate the right strategy, or
- The particular p realization facing the receiver was not one for which the produced strategy gave the correct recommendation.

Our second desideratum essentially states that the probability of both of these events is small; both converge to 0 at a rate that is exponential in the amount of data available:

Definition 2 (Shalev-Shwartz and Ben-David (2014))**.** *A strategy set Γ is PAC-Learnable if there exists an algorithm τ following inequality holds:*

$$\lim_{t\to\infty} \mathbb{P}_{D_t}[\mathbb{P}_{p\sim\sigma}[\tau(D_t)(p) \neq y(p) \mid D_t] < \epsilon] > 1-\delta,$$

for any $y : \mathcal{P} \to A$, $\sigma \in \Delta(\mathcal{P})$, $\epsilon \in [0, 1/2)$ and $\delta \in [0, 1)$. In this case, the algorithm τ is a PAC learning algorithm. We say Γ is efficient PAC-learnable if (1) there exists a polynomial in $1/\varepsilon$ and $1/\delta$, say $T(1/\varepsilon, 1/\delta)$, such that whenever $t \geq T(1/\varepsilon, 1/\delta)$, the previous condition holds, and (2) the number of computation states of the algorith τ is polynomial as well.

According to this definition, the output of the algorithm would be "approximately" correct—in that it gives the optimal prediction—with high probability, given suffficient data. The efficiency part of the previous definition essentially provides a particular (polynomial) rate on the amount of data necessary for an algorithm to provide a good answer.

We emphasize that, the *realized* convergence rate of the algorithm will depend only on what the sender does on-path. While we seek some guarantee for every possible $\pi \in \Pi \subset \Delta(\Theta)$, the particular rate of convergence may depend on the problem (and, of course, endogeneity of σ). The fact that the dataset $(D_t)_{t\in\mathbb{N}}$ will depend on algorithm capabilities makes the realized rate of the algorithm endogenously determined, a distinguishing feature of our exercise. However, notice that whether or not the PAC-condition holds does *not* depend on endogenous variables—this is because we require the inequality to hold *conditional on the data generating process, for all data*

[18] For more background on this notion, see Shalev-Shwartz and Ben-David (2014), for instance.

generating processes—and, therefore, all possible σ^R (in other words, *all* possible sender choices, regardless of whether or not they are chosen).[19]

Along these lines, a useful observation for our subsequent analysis is that the PAC requirement essentially strengthens rationality—that is, PAC learnability is a sufficient condition for the algorithm game to have a Nash equilibrium which approximates $(\sigma^R, y^R(\sigma^R, p))$:

Proposition 4. *If τ is a PAC learning algorithm, the sender's optimal strategy is σ^R and the receiver's long run payoff is as if $y^R(\sigma^R, p)$ were played in every period.*

Proof. If τ is a PAC learning algorithm, then the receiver learns σ accurately in the long run, for any possible sender choice σ. Thus, the long run average expected payoff of the sender is

$$\mathcal{U}(\sigma, \tau) = \mathbf{E}_\theta u(\theta, \sigma, y^R(\sigma, \sigma(\theta)))$$

By the definition,

$$\sigma^R \in \arg\max \mathbf{E}_\theta u(\theta, \sigma, y^R(\sigma, \sigma(\theta))).$$

By PAC learnability,

$$\lim_{t\to\infty} \mathbb{P}[\tau(D_t)(p) \neq y^R(\sigma^R, p)] = 0,$$

implying that $\mathbb{E}[v(\theta_t, p_t, a_t)] \to \mathbb{E}[v(\theta_t, p_t, y^R(\sigma^R, p)]$ as $t \to \infty$; the implication holds due to the assumption that π has finite support, meaning that the payoff is bounded. This implies the long run payoffs are equal to those obtained against a rational player, as desired. □

3.2. Computational Constraints

We now describe the computational aspects of the algorithm's problem, and therefore provide additional details on how $\mathcal{T}$ may be constrained. To motivate this, suppose that we have a computer program which can solve the following problem, for some fixed set of strategies $\mathcal{H}$, an arbitrary distribution d over $\mathcal{P}$, and arbitrary function $y : \mathcal{P} \to A$:

$$\max_{h\in\mathcal{H}} \quad \sum_p \mathbf{1}[h(p) = y(p)]d(p), \tag{3.1}$$

We refer to this step as finding the *best fitting strategy*, and treat it as a black box.

The larger the set $\mathcal{H}$, the harder it is to solve this maximization problem. The crux behind our notion of computational simplicity is that we desire the set $\mathcal{H}$ to be minimal—it does contain

[19] Notice also this definition assumes it is possible to actually approximate $y^R(\sigma, p)$ using some element from Γ. A related concept which does not use this assumption is *agnostic learnability*; as this plays no role in our analysis, however, we do not discuss it further, but see Shalev-Shwartz and Ben-David (2014) for more details.

enough strategies to find a rational reply, but also does not contain more strategies than it needs. Put differently, our notion of complexity is one whereby an algorithm is more complex if it has the ability to solve more difficult versions of (3.1), where the added difficulty emerges due to more possible candidates to check.

Recall that a distinguishing feature of our sender-receiver setting is that it features strategies which require the receiver to reply following many possible actions of the sender. The settings may also feature special structure (e.g., single-crossing payoffs) which dramatically simplify the set of rational replies (e.g., to be monotone). Nevertheless, our running example shows that simply finding the best-fitting strategy within $\mathcal{H}^R$, the set of rational replies, may not lead to rational play. In general, it will be necessary to expand the class of replies considered. Does this mean we will need to assume our computer program will be able to solve harder versions of (3.1)? It turns out the answer to this question is no, as our main results below show.

In asking whether an algorithm can achieve rationality, therefore, we consider the following additional capabilities which we discuss in the rest of the paper:

- Constructing labels based on observations,
- Creating strategies derived from solutions to the above maximization,
- Changing observations of p_t to $\hat{p}_t$.

There are two particular components to the computational simplicity of the algorithm we propose:

3.2.1. Computational Simplicity, Part 1: Using a small set $\mathcal{H}$.

Solving the problem (3.1) gets computationally more demanding the larger the set $\mathcal{H}$ is. Therefore, our goal will be to make this set as simple as possible.

Given that the set of possible sender strategies lies in a subset of $\mathbb{R}^n$, the simplest set of strategies we will generally be able to consider is the set of *hyperplane strategies*. Specifically, let $\mathsf{H}(\lambda, \omega)$ be a hyperplane in $\mathbb{R}^n$: $\exists \lambda \in \mathbb{R}^n$ and $\omega \in \mathbb{R}$ such that

$$\mathsf{H}(\lambda, \omega) = \{p \in \mathbb{R}^n \ : \ \lambda p = \omega\}.$$

Define $\mathsf{H}_+(\lambda, \omega)$ as the closed half space above $\mathsf{H}(\lambda, \omega)$:

$$\mathsf{H}_+(\lambda, \omega) = \{p \in \mathbb{R}^n \ : \ \lambda p \geq \omega\}.$$

Definition 3. *A single threshold (linear) strategy is a mapping*

$$h : \mathcal{P} \to A$$

where $\exists a_+, a_- \in A$, $\lambda \in \mathbb{R}^n$ *and* $\omega \in \mathbb{R}$ *such that*

$$h(p) = \begin{cases} a_+ & \textit{if } p \in \mathsf{H}_+(\lambda, \omega) \\ a_- & \textit{if } p \notin \mathsf{H}_+(\lambda, \omega). \end{cases}$$

These strategies essentially generalize the single-threshold strategies discussed in the context of our running example in Sections 2.1.3 and 2.2.3. For environments like this one, this is the *simplest class of strategies* whereby it will be possible to approximate the rational reply. Notice that, provided the utility functions satisfy an appropriate notion of single-crossing given the particular environment, increasing single-threshold strategies will be optimal. The fact that we also will require "decreasing" single-threshold strategies is perhaps less obvious; however, as we discuss below, in order for our algorithm to work, we require *closedness under action permutation*,[20] which is satisfied by this strategy set.

3.2.2. Computational Simplicity, Part 2: Recursive Updating

Given the fact that (1) the set $\mathcal{H}$ used to solve (3.1) is restricted, and (2) we already know from our discussion of Rubinstein (1993) that such restrictions may prevent the rational benchmark from emerging, it is natural to wonder how we could possibly hope to obtain our promised result on the ability of algorithms to approximate rationality. The answer is that we will allow the algorithms to construct new decision rules based on the solutions to (3.1). We now describe these capabilities in detail. Specifically, we consider the case where $\mathcal{T}$ is restricted to emerge as the outcome of an *ensemble algorithm.*

Definition 4. *Strategy* H *is an ensemble of* $\mathcal{H}$ *if* $\exists h_1, \ldots, h_K \in \mathcal{H}$ *and* $\alpha_1, \ldots, \alpha_K \geq 0$ *such that*

$$H(\sigma, p) = \arg\max_{a} \sum_{k=1}^{K} \alpha_k \mathbf{1}[a = h_k(\sigma, p)]$$

We say that an algorithm is an ensemble algorithm if it produces an ensemble of $\mathcal{H}$*.*

Without loss of generality, we can assume that $\sum_{k=1}^{K} \alpha_k = 1$, since if not we can simply divide by this sum and obtain the same strategy. We can interpret H as a weighted majority vote of $h_1, \ldots, h_K$. An ensemble algorithm constructs a strategy through a linear combination of elements of $\mathcal{H}$. Since the final strategy is constructed through a basic arithmetic operation, one can easily construct an more elaborate one from rudimentary strategies. Ensemble algorithms have been remarkably successful in real world applications (Dietterich (2000)).

[20] More formally, this requirement states that if $G : A \to A$ is a permutation, and $h(p) \in \mathcal{H}$, then $G(h(p)) \in \mathcal{H}$.

How can the ability to produce ensembles be leveraged, given only the ability to solve (3.1) for limited $\mathcal{H}$? In our proofs below, we first will specify a probability distribution $d_0(p)$ over observables, and then use this to find a best fitting hypothesis. In subsequent steps, however, the algorithm will:

- Find a best fitting strategy—i.e., solve (3.1), using $d_k(p)$.
- Update the distribution in (3.1), say from $d_k(p)$ to $d_{k+1}(p)$, and
- Find new weights $\{\alpha_j\}_{1\leq j\leq k}$ to place on each strategy.

Definition 5. *An algorithm τ is a* ***recursive ensemble algorithm*** *using $\mathcal{H}$ if:*

- *The final strategy produced is an ensemble of $\mathcal{H}$, and*
- *Each strategy $h_k \in \mathcal{H}$ is found by solving (3.1), using some distribution $d_k(p) \in \Delta(\mathcal{P})$, and*
- *Either $k = 1$, or $\{d_k(p)\}_{p\in\mathcal{P}}$ is a function of $y^R(\sigma, \cdot), h_{k-1}, \alpha_{k-1},$ and $\{d_{k-1}(p)\}_{p\in\mathcal{P}}$ alone.*

4. Statements of the Main Results

We now present our main results, exhibiting an equilibrium of the algorithm game where the rational reply is emulated. The algorithm we highlight is computationally simple, in the sense previously described, and achieves a PAC guarantee as well.

Proposition 5. *Suppose the values of $y^R(\sigma, p)$ are observed by the algorithm, $\forall(\sigma, p)$. Then there exists an algorithm which*

- *Is a recursive ensemble algorithm,*
- *Uses $\mathcal{H}$ equal to the set of single threshold classifiers, and*
- *achieves a PAC-guarantee,*

Denoting this algorithm by τ_A, (σ^R, τ_A) thereby forms a Nash equilibrium of the algorithm game, which emulates (σ^R, y^R).

It follows that letting $\mathcal{T}$ contain recursive ensemble algorithms is enough to emulate rationality. To interpret the significance of this result, let us revisit the running example. In this case, $\mathcal{H}^R$ is (a subset of) the set of single threshold classifiers, and as we have argued, simply being able to solve (3.1) is not sufficient to emulate rationality. Our result shows that despite this, it is not necessary to solve a problem more computationally demanding than (3.1)—all that is required is an appropriate specification of the algorithm, as we do below.

We describe the key property on the set of baseline strategyes necessary for this algorithm to work; this property is known as *weak learnability*. In our proofs, we identify *closure under permutations* as the key property which generally is necessary in order for this property to be satisfied. Therefore, even if $\mathcal{H}^R$ only includes increasing threshold strategies, it is necessary for $\mathcal{H}$ to include both increasing and decreasing single-threshold strategies.

An important assumption above is that the labels $y^R(\sigma, p)$ are observable (i.e., part of the data observed by the algorithm). If they were not, then the algorithm would need to perform the added step of inferring them. This will yield an algorithm $\tau_{\hat{A}}$, which coincides with τ_A with the added step of inferring the rational replies. This requires a slightly stronger assumption on the stage games, but nevertheless we can still show that we obtain an analogous result:

Proposition 6. *Suppose that $y^R(\sigma, p)$ is a strict best response $\forall \sigma$, and that D_t contains θ_t (or equivalently, the payoffs following each possible action of the receiver). Then there exists an algorithm which*

- *Is a recursive ensemble algorithm,*
- *Uses $\mathcal{H}$ equal to the set of single threshold classifiers, and*
- *achieves a PAC-guarantee,*

Denoting this algorithm by $\tau_{\hat{A}}$, $(\sigma^R, \tau_{\hat{A}})$ thereby forms a Nash equilibrium of the algorithm game, which emulates (σ^R, y^R).

The strict best response condition ensures that the unique value of $y^R(\sigma, p)$ *must* be learned in order to achieve a strict best reply. In the applications we study below, we verify that it indeed does hold. If multiple values of $y^R(\sigma, p)$ are optimal, then it may be that the sender deviates from σ in order to break the receiver's indifference in a particular way. The following example illustrates the difficulty that might emerge if this assumption does not hold, and why an improvement is likely not possible:

Example 1. *Consider the following version of the ultimatum game, and let us take for granted that the receiver uses an algorithm which computes an optimal reply to the empirical distribution (as we show exists in our proposal). The sender offers the receiver a payment, say $\mathcal{P} = \{0, 1/n, \ldots, 1\}$. The receiver can accept ($a = 1$) or reject ($a = 0$) the offer. The state θ is equally likely to be -1 or 1. Sender's payoff is $u(\theta, p, a) = 1 - p$, while receiver's payoff is $v(\theta, p, 0) = 0$ and $v(\theta, p, 1) = \theta + p$.*

When the distribution over θ is known, the subgame perfect equilibrium of this game involves the sender choosing $p = 0$ with probability 1, and the receiver choosing to accept the offer. However, the receiver's payoff would be unchanged when rejecting the offer, since the rational receiver would be indifferent. Clearly, this is the unique subgame perfect equilibrium outcome.

When the distribution over θ is not known, however, notice that for all finite time horizons, it is equally likely that there are more $\theta = 1$ observations than $\theta = -1$ observations. Therefore, in expectation, the receiver will choose $a = 1$ half the time and $a = -1$ half the time, yielding long run payoff to the sender of $1/2$.

In this case, the sender would have a strictly profitable deviation—by choosing $p = 1/10$ with probability 1, given a sufficiently large amount of data, the receiver will find it strictly optimal to choose $a = 1$ (eventually). Therefore, the sender obtains long run payoff of $9/10$, strictly larger than what they obtain by playing σ^R.

More generally, the example shows that whenever there are multiple best replies, subgame perfection often might require that one of them is never taken in the case of indifference, in order to induce particular behavior from the first mover. As the example shows, this could lead to quite a different outcome when the distribution over θ needs to be estimated.

Before we turn to some additional applications and an overview of the proofs of the main results, we briefly comment that these results actually understate how quickly convergence may be obtained. For instance, in our running example, σ^R is a deterministic function. For this reason, however, it is reaassuring that we do *not* need to make the set $\mathcal{H}$ in (3.1) excessively large, even though Section 2.2.3 suggested that we might. In other words, an attractive property of our algorithm is that it *minimizes* the set of baseline strategies. This is in contrast to the conclusions of Propositions 1 and 2, which suggested that richer strategies were necessary to achieve rationality. While this is true, it can be achieved without increasing computational demands (as per (3.1)).

5. Applications

Before turning to the proofs of these results, we describe some other interactions which our model speaks to. This will articulate the added richness relative to the Rubinstein (1993) setting, and also show the usefulness of our results. Indeed, in order for our algorithm to work, the only necessary condition is on $y^R(\sigma, p)$ being a best reply. In these cases, if σ is simple (e.g., cosnists of only a single observation), then the algorithm will achieve a much better rate relative to when σ is arbitrary.

5.1. Wrapping up the Running Example

We briefly record that the strict best response condition is satisfied in the monopoly market example in Sections 2.1.3 and 2.2.3.

Lemma 1. *Consider the monopoly market model as specified by Rubinstein (1993).*[21] *Fix σ which assigns $p > v_L$ with positive probability, satisfying*

$$\mathbf{E}_\theta v(\theta, p, 1) \geq 0. \tag{5.2}$$

Then, the ex ante expected profit of the principal against $\tau_{\hat{A}}$ from σ is strictly smaller than from σ':

$$\mathcal{U}(\sigma^R, \tau_{\hat{A}}) > \mathcal{U}(\sigma, \tau_{\hat{A}}).$$

This Lemma shows that, given the algorithm $\tau_{\hat{A}}$, the outcome of the interaction will emulate the rational outcome, overturning the implication of Rubinstein (1993). The proof essentially amounts to checking the strict best response condition of Proposition 6.

5.2. Labor Markets

For an application with more than two possible actions, we us consider a labor market signaling model. Here, the receiver takes the role of the firm and the sender takes the role of the worker from the Spence signalling model (see also Maskin and Tirole (1992)). The true state is the productivity of the worker $\theta \in \Theta = \{H, L\}$. Conditioned on θ, a worker chooses p which we interpret as education level. Her strategy is

$$\sigma : \Theta \to \mathcal{P} \subset \mathbb{R}_+.$$

The payoff function of the sender is

$$u(\theta, p, a) = a - \frac{p}{\theta + 1}$$

We abstract away the competition among multiple firms in the labor market. Conditioned on p, the labor market wage is determined according to the expected productivity $\mathbf{E}(\theta : p)$ conditioned on p. The firm has to pay the worker the equal amount of the expected productivity because of (un-modeled) competition among firms. The receiver's goal is to make an accurate forecast about

[21] This proof uses the precise payoff specification of Rubinstein (1993), which our prior exposition does not include. For completeness, this can be found in the Appendix.

the expected productivity of the worker. The payoff of the receiver is

$$v(\theta, p, a) = -(\theta - a)^2$$

If the support of $\sigma(p : H)$ is disjoint from the support of $\sigma(p : L)$, σ is a separating strategy. If a separating strategy is an equilibrium strategy, then the equilibrium is called a separating equilibrium. We often focus on the Riley outcome, which maximizes the ex ante expected payoff of the principal among all separating equilibria.

5.2.1. Analysis

The firm seeks to predict worker quality. Hence if A is a real line, then

$$y^R(\sigma, p) = \arg\max_{a \in A} \mathbf{E}_\theta \left[v(\theta, p, a) : p, \sigma \right]$$

where the posterior distribution over θ is calculated via Bayes rule from σ and the prior over θ. Strict concavity of v implies that $y^R(\sigma, p)$ is a strict best response $\forall \sigma, p$.

Since there are multiple actions in this example, single threshold decision rules are parameterized by (a^+, a^-, p^0):

$$h(p) = \begin{cases} a^+ & \text{if } p \geq p^0 \\ a^- & \text{if } p < p^0. \end{cases}$$

In each round, h_t solves

$$\max_{h \in \mathcal{H}} \mathbf{E}_\theta \left[v(\theta, p, a) : p, \sigma \right]$$

if the data includes σ. We construct τ_A accordingly. If σ is not observable by the algorithm, we estimate the posterior distribution of σ conditioned on each p to construct $\tau_{\hat{A}}$. If the agent learns $y^R(\sigma, p)$ eventually $\forall \sigma, p$, then the principal's choice σ^R

$$\mathbf{E}[u(\theta, p, y^R(\sigma, p)) : \sigma, p] = \sum_\theta \sum_p u(\theta, p, y^R(\sigma, p)) \sigma(p : \theta) \pi(\theta).$$

If σ^R entails separation by the high productivity worker, then the Riley outcome is the solution, that generates the largest ex ante expected surplus for the principal among all separating equilibria. In order to satisfy the incentive constraint among different types of the principal, the principal with $\theta = H$ incurs the signaling cost. If the signaling cost outweighs the benefit of separation, then σ^R is the pooling equilibrium where both types of the workers takes the minimal signal.

Proposition 6 requires $y^R(\sigma, p)$ to be a strict best response $\forall \sigma, p$. In this example,$|A| = J < \infty$,

$y^R(\sigma, p)$ may not be a strict best response for some σ and p. Let us assume that

$$A = \{a_0 = 0, a_1, \dots, a_J\}$$

and $a_i - a_{i-1} = \Delta > 0$ and $a_F = 1 + \sup \Theta > 0$. Although $y^R(\sigma, p)$ may not be a strict best response for some σ and p, the set of best responses contains at most 2 elements, which differ by $\Delta > 0$. Abusing notation, let $y^R(\sigma, p)$ be the set of best responses, if the agent has multiple best responses at p. Applying the convergence result, we have $\exists T$ such that, $\forall t \geq T$,

$$\mathbb{P}\left(\exists y \in y^R(\sigma, p), y = \tau_{\hat{A}}(D_t)(p)\right) < e^{-\rho t}.$$

For a sufficiently small $\Delta > 0$, σ^R is either a strategy close to the Riley outcome, or the pooling equilibrium where both types of the principals choose the smallest value of p.

5.3. Insurance

The following is borrowed from Maskin and Tirole (1992). Suppose that the principal (sender) is a shipping company seeking to purchase insurance from an insurance company, an agent (receiver) that is seeking to delegate the decision of whether to offer the terms put forth by the shipping company. The principal seeks insurance every period, but faces risk (e.g., due to the location of shipping demand) that is idiosyncratic every period.

In this case, we imagine the principal choose terms within some compact set $\mathcal{P} \subset \mathbb{R}^2$, where $p = (x, q)$ denotes a policy which provides a payment x in the event of a loss, and costs an amount q. If $\theta \in \{L, H\}$ (with $L < H$) denotes the probability of a loss, then the principal's utility is:

$$u(\theta, p, a) = \begin{cases} (1-\theta) f(I - q) + \theta f(I - q - L + x) & a = 1 \\ (1-\theta) f(I) + \theta f(I - L) & a = -1 \end{cases},$$

for some concave f. The agent's utility is:

$$v(\theta, p, a) = \begin{cases} q - \theta x & a = 1 \\ 0 & a = -1 \end{cases}$$

It is natural to consider $\mathcal{P}$ whereby, against a rational buyer, the principal would seek a high level of insurance when risk is high (i.e., $\theta = H$), and avoid insurance when risk is low (i.e., $\theta = L$). In contrast, the agent's payoff may be decreasing in the quantity of insurance when $\theta = H$, while increasing in the quantity of insurance when $\theta = L$.

We now verify the implications of this observation on the sender behavior in our particular examples, showing that this results in the sender-preferred Stackleberg outcome is emerging. This requires us to verify the previously discussed conditions in the context of these applications.

5.3.1. Analysis

The decision problem of the agent is to identify each pair (x, q) of payment x and cost q as an acceptable constract $(a = 1)$ or not $(a = -1)$. The added difficulty in this example is that the action space is multidimensional. However, we can still specify a "baseline strategy space" to be the set of single-threshold classifiers. That is, as per our algorithm, we assume access to strategies induced by hyperplanes:

$$\mathsf{H}(\lambda_x, \lambda_q, \omega) = \{(x, q) : \lambda_x x + \lambda_q q = \omega\}$$

and then take:

$$h(x, q) = \begin{cases} 1 & \text{if } (x, q) \in \mathsf{H}^+(\lambda_x, \lambda_q, \omega) \\ -1 & \text{otherwise.} \end{cases}$$

We can construct $\tau_{\hat{A}}$ by estimating $\mathbf{E}v(\theta, p, a)$ for each (p, a).

To apply Proposition 8, it is necessary to show the strict best reply condition. This turns out to hold in our setting:

Lemma 2. *Suppose that σ assigns a positive probability to (x, q) where*

$$\mathbf{E}(q - \theta x : (q, x)) = 0$$

for $x > 0$. Then σ is not a best response to $\tau_{\hat{A}}$.

Following the same logic as in the previous example, we conclude that if σ is a best response to $\tau_{\hat{A}}$, then

$$\tau_{\hat{A}}(D_t)(q, x) = y^R(\sigma, (x, q))$$

with probability 1. A best reply σ to $\tau_{\hat{A}}$ emulates $(\sigma^R, y^R(\sigma^R, p))$.

6. Overview of the Proofs of the Main Results

This section discusses the techniques used to prove the results from the previous section, with details being relegated to the Appendix. There are two parts to the proof. The first part is to specify the algorithm so that, despite limitations of $\mathcal{H}^R$, a recursive algorithm can be specified to approximate the rational reply. This is sufficient to prove Proposition 5. To prove Proposition 6, we additionally use results from large deviations theory to derive similar exponential bounds at which the algorithm can infer the best reply to a given observed sender choice. As a result, the PAC condition will still hold.

6.1. Specifying the Algorithm with observed $y^R(\sigma, p)$ (Proposition 5)

6.1.1. Weak Learnability

The sufficient condition which ensures we can approximate an arbitrary decision rule combining single-thresholds is *weak learnability.* Roughly speaking, weak learnability says that the optimally chosen strategy outperforms someone who had some very minimal knowledge of the truth of the hypothesis. That is, it must be that using only the strategy set, one can do better than someone who made a random guess, provided this guess would be made correct with some arbitrarily small probability. While this may seem permissive—and indeed, it is certainly less stringent than requiring the ability to approximate the truth with high probability—the difficulty in achieving it is the fact that this guarantee must be uniform over all possible distributions.

We formally define this as follows:

Definition 6. *If $|A| = 2$, a strategy set $\mathcal{H}$ is weakly learnable if, for every distribution d over observations $p \in P(\sigma)$ and labels $y(p)$, the optimal weak hypothesis satisfies:*

$$\sum_{p \in P(\sigma)} D(p)(\mathbf{1}[y(\sigma, p) \neq h(p)] - \mathbf{1}[y(\sigma, p) = h(p)]) \geq \rho.$$

If $|A| > 2$, a strategy set $\mathcal{H}$ is weakly learnable if, for every distribution d over observations $p \in P(\sigma)$ and labels $y(p)$, the optimal weak hypothesis satisfies:

$$\sum_{p \in P(\sigma)} \mathbf{1}[\overline{h}(p) \neq y(p)]d(p) \leq \sum_{p \in P(\sigma)} \mathbb{E}_{\tilde{y} \sim B}[(1 - \rho)\mathbf{1}[\tilde{y} \neq y(p)]]d(p),$$

for some $\rho > 0$ and some distribution B over A.

The second condition is a generalization of the first, though the first is perhaps more familiar from the machine learning literature (as most attention has focused on the case where there are at most two possible choices). This condition reflects the idea that the strategy randomly guesses the label according to some distribution B, but is "flipped to being correct" with probability ρ. For the $|A| > 2$ case, the right hand side describes the expected error in such a case, and the left hand side describes the error from the optimally chosen element of $\mathcal{H}$.

If weak learnability fails, then *no* recursive ensemble algorithm can be built to approximate

$y(p)$ based on $\mathcal{H}$ alone.[22] Perhaps more surprising is that it is tight, a fact which we discuss further in Section 6.3.1. For now, we simply mention that the set of single threshold classifiers is weakly learnable.

Proposition 7. *The set of single-threshold classifiers satisfies the weak learnability condition of Definition 6.*

Proof. See Appendix A. □

The proof of the Appendix actually proves a stronger theorem which may be of independent interest, and a version of it is used later when we consider the$|A| > 2$ case as well: Any hypothesis class that *contains all label permutations* can at least match the random guess guarantee. The proof of this intermediate lemma uses a duality argument in order to show that no distribution can lead to a lower payoff when this condition is satisfied. Importantly, however, this is true for *any* hypothesis class, including the trivial one. This observation allows us to show that the added richness of single-threshold classifiers is sufficient to provide the additional gain over random guessing.

6.1.2. From Weak Learnability to Decision Rules

We now describe the algorithm which achieves our desired guarnatee; the Appendix shows that this algorithm is a valid choice for Proposition 5. Recall that for this result, we assume $y^R(\sigma, p)$ is in $\mathcal{D}$. In addition, for simplicity we now only focus on the case where

$$A = \{-1, 1\}$$

leaving the general case to the appendix. For this special case, the specification of the algorithm parameters coincides with the Adaptive Boosting algorithm τ_A of Schapire and Freund (2012). We now describe precisely how this algorithm works.

To start the algorithm, we require only an initialization of d_1, which we will use in solving (3.1)—we will take this to be the uniform distribution over $\mathcal{P}$:

1. Given a distribution d_k over $\mathcal{P}$, Let $h_k(p)$ be the strategy which solves (3.1), and define

$$\epsilon_k = \mathbb{P}_{d_k}\left(h_k(p) \neq y^R(\sigma, p)\right) \tag{6.3}$$

[22] For example, imagine $\mathcal{H}$ only consists of trivial strategies (i.e., those that always choose a single action). A corollary of a result in Appendix 3 is that these strategies can do equally well as a random guesser. However, it is clear that they cannot do strictly better, as they are restricted to giving the same guess to all possible p, unlike a random guesser who is correct with an added probability ρ.

as the probability that the optimal strategy h_k at k misclassifies p under d_k.

2. If $\epsilon_k = 0$, then the algorithm has returned a perfect forecast, and thus the algorithm will return output h_k as the final forecasting rule.

3. Otherwise, $\epsilon_k > 0$, and thus set:

$$\alpha_k = \frac{1}{2} \log \frac{1 - \epsilon_k}{\epsilon_k} \tag{6.4}$$

4. Finally, for each p in the support of σ,

$$d_{k+1}(p) = \frac{d_k(p) \exp(-\alpha_k y^R(\sigma, p) h_k(p))}{Z_k}$$

where

$$Z_k = \sum_p d_k(p) \exp(-\alpha_k y^R(\sigma, p) h_k(p)).$$

Provided the algorithm never terminates at step 2 above, then it terminates at some fixed K, specified in advance, and the final receiver strategy produced by the algorithm is:

$$\tau_A(D_k)(p) = \arg\max_{a \in A} \sum_{t=1}^{k} \alpha_t \mathbf{1}(h_t(p) = a).$$

Following Schapire and Freund (2012), we can show that

$$\mathbb{P}\left(\tau_A(D_t)(p) = y^R(\sigma, p)\right) \geq 1 - e^{-t\rho(G)} \tag{6.5}$$

for any mixed strategy σ, where ρ is some function (determined in the proof) and G is the number of elements in the support of σ.[23]

6.2. Inferring $y^R(\sigma, p)$ from $\{v(\theta_t, p_t, a)\}$ (Proposition 6)

Next, we drop the assumption that the receiver observes $y^R(\sigma, p)$, and instead uses an estimate, which we denote by $\hat{y}_t(p)$. This modification introduces the possibility that $\hat{y}_t(p) \neq y^R(\sigma, p)$—the machine learning literature refers to this possibility as *classification noise* or *label noise*. Boosting algorithms like the ones we propose using are known to perform quite poorly in the presence of classification noise. While the reasons for this are somewhat technical, very briefly they relate to

[23] A sketch of the proof is in Appendix B.

how the particular coefficients α_k above are chosen. Notice that ϵ_k in (6.3) requires the $y^R(\sigma, p)$ to be observed. Without label noise, the choices of α_k specified above emerge from minimizing a particular convex function of the ϵ_ks which achieve the same minimum as the (non-convex) objective function that the algorithm would like to minimize. When label noise is present, this replacement function itself need not be convex (see, for instance, Long and Servedio (2010) or Freund (2009); Frenay and Verleysen (2014) provides a survey of the issue and proposed solutions).

What is our solution, and why does it get around this problem? In the papers referenced above, label noise is modelled as the possibility that there is some fixed probability ρ according to which the correct action is not chosen. By contrast, in our setting, label noise emerges because the algorithm would like to find the action a which maximizes:

$$\sum_\theta v(\theta, p, a)\pi(\theta : p) \tag{6.6}$$

where π is computed via Bayes rule. A natural proposal is to simply replace π by its sample analog, and indeed, this will be our proposal as well.. Our proposed algorithm, $\tau_{\hat{A}}$ coincides with the one in the previous section, but where first $\hat{y}_t$ is estimated and $y^R(\sigma, p)$ is replaced with $\hat{y}_t(p)$. More precisely, we make three changes:

- Replace $y^R(\sigma, p)$ with $\hat{y}_t(p) = \arg\max \sum_\theta v(\theta, p, a)\hat{\pi}_t(\theta : p)$, where $\hat{\pi}_t(\theta : p)$ denotes the fraction of times state θ is observed when the sender's choice was p,
- Choose h_k to solve:

  $$\max_{h \in \mathcal{H}} \sum_p h(p) d_t(p)[1 \cdot f_t^y(p) - 1 \cdot (1 - f_t^y(p))]$$

  instead of (3.1), where f^y is the empirical probability that $\hat{y}_t(p) = 1$ at the beginning of period t. Thus, $\hat{y}_t(p) = -1$ with probability $1 - f_t^y(p)$.
- Replace the errors ϵ_t with $\hat{\epsilon}_t$, where

  $$\hat{\epsilon}_t = \sum_p d_t(p) \left[f_t^y(p)\mathbf{1}(h(p) = 1) + (1 - f_t^y(p))\mathbf{1}(h(p) = -1) \right].$$

The key feature that distinguishes our setting from other settings with classification noise is that for us the noise vanishes in the limit as $t \to \infty$. Of course, this property is only sensible in our setting due to the *origin* of the classification noise—namely, that the sender might be randomizing their strategy, and making the ex-post payoff following some p possibly different from the expected payoff following p. Now, the observation that label noise vanishes by itself is not enough to eliminate this problem. Our goal is a PAC-guarantee, which requires not just that the noise in $\hat{y}_t(p)$ vanishes, but also that this vanishes quickly, *uniformly* over p. For our modification to work, $\hat{y}_t(p)$ must satisfy the large deviation property (LDP):

Definition 7. *$\hat{y}_t(p)$ satisfies large deviation properties (LDP) if $\exists \lambda > 0$ such that, $\forall p$ in the support of σ,*

$$\limsup_{t\to\infty} -\frac{1}{t}\log \mathbb{P}\left(y^R(\sigma,p) \neq \hat{y}_t(p)\right) \leq \lambda. \tag{6.7}$$

If an estimator satisfies LDP, the tail portion of the forecating error vanishes at an exponential rate, as the sample average of i.i.d. random variables converges to the population mean. If an estimator fails to satisfy LDP, the finite sample property of the estimator tends to be extremely erratic (Meyn (2007)). Most estimators in economics satisfy LDP.

With $\tau_{\hat{A}}$, we can construct labels from data, and that for the hypothesis class of interest the weak learnability condition is satisfied. The last step to show the algorithm works, in the case where the set of possible p has finite support, is that the output of the algorithm will indeed converge to the rational reply, as dictated by the labels, provided the weights are specified correctly.

Proposition 8. *Suppose that $\hat{y}_t$ satisfies uniform LDP and that $y^R(\sigma,p)$ is a strict best response $\forall p$. Then, $\forall \sigma$ that randomizes over G elements of $\mathcal{P}$, $\exists T$ and $\exists \rho(G) > 0$ such that*

$$\mathbb{P}\left(\tau_{\hat{A}}(D_t)(p) = y^R(\sigma,p)\ \forall t \geq T\right) \geq 1 - e^{-t\rho(G)}.$$

Proof. See Appendix B. □

6.3. Discussion of Other Results

6.3.1. Specifying the Algorithm with Multiple Actions

In Section 6.1.2, we used the Adaptive Boosting algorithm, as introduced by Schapire and Freund (2012), to specify the α_k weights and the updates if $|A| = 2$. The original Adaptive Boosting algorithm only applies to the case of $|A| = 2$. To handle the case of $|A| > 2$, we appeal to a generalization introduced by Mukherjee and Schapire (2013). The basic idea behind this algorithm is largely the same, with one minor drawback, which is that algorithm depends on the constant ρ in Definition 6, and thus this must be computed in advance. While our work shows an algorithm exists, the computation of the learnability constant is more indirect and hence explicitly finding a parameter that works is more difficult. The arguments for these proofs follow from results in the machine learning literature (see Schapire and Freund (2012)), which we can apply to show that this algorithm can yield a response for which the misclassification probability vanishes.

As for Proposition 8, its proof is stated for the general case. The proof reveals that the rate at which the probability of misclassfication vanishes is determined entirely by the number of sender actions in the support of σ. Thus, the algorithm is efficient (in the sense of Definition 2) in that it maintains an exponential rate of convergence (Shalev-Shwartz and Ben-David (2014)).

6.3.2. Accommodating Richer Sender Action Spaces

While $\tau_{\hat{A}}$ is designed to be robust against parametric details of the underlying problems, the algorithm is still vulnerable to strategic manipulation by the rational sender. The proof of Proposition 8 reveals that the rate of convergence is decreasing as the number of sender actions in the support of σ increases. The sender can randomize over infinitely many messages to slow down the convergence rate arbitrarily. That said, such manipulation would be short lived, and therefore have limited gains. Nevertheless, in order to ensure that there are only a finite number of observations that the algorithm may observe, it is necessary to augment the observation space so that the distribution facing the receiver can be treated as discrete.

In the appendix, we describe two modifications which allow this to be done. Both involve modifying the price space so that observations of p_t are replaced by $\hat{p}_t$. The first involves "discretizing" the price space, so that each price is instead replaced by an interval of prices, with the same optimal action being taken in each interval. The second involves adding "noise" to the prices; this process allows us to show that the "noisy" prices inherit sufficient smoothness (at the cost of accuracy, of course) so that the problem of estimating actions is well behaved.

7. Conclusion

This paper has studied an algorithm that can approximate rational behavior in settings where a first mover's action can convey information about a payoff relevant state. This algorithm emerges as an equilibrium choice in an algorithm game, a framework which allows us to study the incentives to exploit an algorithm and the implications of constraints on algorithm capabilities. To precisely describe the sense in which our proposal approximates rationality, we used the framework of PAC learnability as a metric of algorithm performance. As long as some initial set of strategies satisfy weak learnability, an algorithm can be specified which ensures the receiver takes an optimal response to the sender's action. As noted by Rubinstein (1993), this need not be the case when the receiver's behavior follows from the *optimally* chosen single-threshold classifier given the sender's strategy. However, being able to combine strategies is enough to overcome this limitation, even if it only remains possible to find the "best" strategy from within this limited set.

In settings featuring strategic inference, a rational receiver must be able to update beliefs about an underlying state (thus influencing the optimal response). This adds a complicating feature that the ex-post optimal action is only observed with noise. Generally, this property can inhibit the performance of Adaptive Boosting. Yet because this noise diminishes with the size of the sample, we are still able to show this presents no added difficulty (thanks to results from large deviations theory). As discussed above, if the amount of label noise were bounded away from zero, then our approach need not be successful (a well-known issue with Boosting algorithms). However, in our setting, the fact that label noise vanishes is well-motivated, due to the assumption that data consists of ex-post payoffs (whereas ex-ante payoffs determine optimal replies). This contribution was necessary due to the uncertainty inherent in our applications of interest.

We have sought to articulate the following tradeoff in the design of statistical algorithms to mimic rationality: on the one hand, simply fitting a single-threshold classifier to data will fall short of rational play and be exploited. On the other hand, it may not be clear why this is the end of the story. By adding the ability to fit strategies repeatedly and combining them in particular ways, we show how the rational benchmark can be restored. Here, we have taken as a black box the ability to fit single-threshold classifiers. But given this, our algorithm specifies exactly how to put them together in order to construct a strategy which can mimic rationality arbitrarily well.

We have focused on a simple yet general setting where the comparison to the rational benchmark is most transparent. Still, we believe that many concerns highlighted by the machine learning literature regarding the design of algorithms can speak to issues of interest to economic theorists. Given how productive the machine learning literature has been in terms of designing algorithms for the purposes of classification, we hope that our work will inspire further analysis of how these algorithms behave in strategic settings.

A. Weak Learnability Proofs

The proof of Proposition 7 uses the following Lemma:

Lemma 3. *Let $\mathcal{H}$ be an arbitrary hypothesis class with the property that for every $h \in \mathcal{H}$ and every permutation $\pi : A \to A$, the composition $\pi \circ h$ is contained in $\mathcal{H}$. Then this hypothesis class can do at least as well as a uniform random guesser.*

Proof. Let Π be the set of all possible permutations on A, noting that$|\Pi| = k!$. Fix an arbitrary $h \in \mathcal{H}$, and define $h^\pi = \pi \circ h$. Let $c_{j,y}$ be the cost of assigning label y to price p_j. Define

$$\sum_{\pi \in \Pi} c_{j,h^\pi(p_j)} = \overline{c}_j.$$

In particular, note that this is invariant to the true label of j. As a result, the random guesser's expected payoff on observation j is is $\overline{c}_j/k!$. To see this, note that $h(p_j)$ gives some fixed guess regarding the label of price p_j. Then randomizing over permutations is equivalent to randomizing over labels, as there are an equal number of permutations which flip the label according to $h(p_j)$ and every other label.

We therefore obtain the following matrix equation, for an arbitrary $\rho \in (0, \infty)$, where the number of columns is $k!$ and the number of rows is the number of possible prices.

$$\left(\begin{array}{c|c|c} \begin{matrix} c_{j,h(p)} \\ -\overline{c}_j/k! \end{matrix} & \cdots \quad \cdots & \begin{matrix} c_{j,h^\pi(p)} \\ -\overline{c}_j/k! \end{matrix} \end{array}\right) \cdot \begin{pmatrix} \frac{\rho}{k!} \\ \vdots \\ \frac{\rho}{k!} \end{pmatrix} = \mathbf{0}$$

Also note that:

$$(1/\rho, \cdots, 1/\rho) \cdot \begin{pmatrix} \frac{\rho}{k!} \\ \vdots \\ \frac{\rho}{k!} \end{pmatrix} = 1$$

So as long as $\rho > 0$, by the theorem of the alternative, we therefore cannot have that a vector $\mathbf{x}$ exists with:

$$\left(\begin{array}{cc} \multicolumn{2}{c}{c_{j,h(p)} - \overline{c}_j/k!} \\ \hline \vdots & \vdots \\ \vdots & \vdots \\ \hline \multicolumn{2}{c}{c_{j,h^\pi(p)} - \overline{c}_j/k!} \end{array}\right) \cdot \mathbf{x} \geq \begin{pmatrix} \frac{1}{\rho} \\ \vdots \\ \frac{1}{\rho} \end{pmatrix}.$$

Let $D(p)$ be an arbitrary distribution. Since $\sum_{p \in P} D(p) = 1$, this implies we can find some π such that:

$$\left(\sum_{p_j \in P} D(p_j)(c_{j,h^\pi(p_j)} - \frac{\overline{c}_j}{k!})\right) < \frac{1}{\rho}.$$

Taking $\rho \to \infty$ and rearranging gives:

$$\left(\mathbb{E}_{p\sim D}[c_{j,h^{\pi}(p_j)}]\right) \leq \mathbb{E}_{j\sim D}\left[\frac{\overline{c}_j}{k!}\right]$$

Recalling again that the right hand side of this inequality is the payoff of the random guesser, we have shown that for every possible distribution over prices, we can find some permutation which delivers a cost bounded above by the random guesser. This proves the Lemma. □

Proof of Proposition 7. Let $\mathcal{H}$ be the set of hyperplane classifiers. We prove this by contradiction. If there were no universal lower bound on the error, then we would have, for all ρ, a distribution D_ρ and cost $c^{\rho}_{j,y}$ (without loss normalized to be on the unit sphere themselves) with the property that:

$$\max_{h\in\mathcal{H}} \sum_{p\in P} D_\rho(p) c^{\rho}_{j,h(p)} < U^{\rho}_c,$$

where U^{ρ}_c is the payoff of the uniform random guesser who is correct with added probability ρ. Taking $\rho \to 0$ and passing to a subsequence if necessary, compactness of the unit sphere implies that we can find a distribution D^* and cost function c^* such that:

$$\max_{h\in\mathcal{H}} \sum_{p\in P} D^*(p) c^*_{j,h(p)} = U^0_c,$$

where we note by Lemma 3 that at least this bound can be obtained by permutation the labels if necessary. We will arrive at a contradiction by exhibiting a single-hyerplane classifier that achieves a strictly better accuracy, given D^*. Note that $\mathcal{H}$ contains the set of "trivial" classifiers, which give all menus the same label. Also note that the only non-trivial case to consider is when there are at least two prices in the support of D^*; if there were only one price, then simply choosing the prediction corresponding to the label on that price would yield a perfect fit. Since, by assumption, no classifier does better than random guessing, it must be the case in particular that each trivial classifier cannot exceed the random-guess bound. On the other hand, by our previous result, we know there *does* exist a trivial classifier which achieves at least this bound, for *any* D supported on P.

Let $P = \{p_1, \dots, p_k\}$ be the set of prices supporting D^*, and let $\tilde{p} \in P$ be a price in that is also an extreme point of the convex hull of P. Without loss of generality, assume that $\tilde{p}$ is nontrivial, in the sense that it does not give the same cost to all labels. Note that indeed, this is without loss, since for any such price, the choice of classification is irrelevant.[24] Note that $\tilde{p}$ is not in the convex hull of $P\backslash\{\tilde{p}\}$. Therefore, by the separating hyperplane theorem, we can find an $h \in \mathcal{H}$ which (strictly) separates $\tilde{p}$ from $P\backslash\{\tilde{p}\}$. Denote such a hyperplane by h^*, and note that the set of hyperplane classifiers contains classifiers which assign *any* two labels (possibly the same label) to prices depending on which side of h^* they lie on.

Also note that, again by our previous result, a trivial classifier supported on $P\backslash\{\tilde{p}\}$ can achieve the random guess guaranatee if p is distributed according to the conditional distribution on this set. In other words, our prior lemma implies that there exists $y^* \in A$ such that:

$$\sum_{p_j\in P\backslash\tilde{p}} \frac{D^*(p_j)}{\sum_{q\in P\backslash\tilde{P}} D^*(q)} c^*_{j,y^*} = U^0_{c^*}.$$

On the other hand, a classifier which separates $p_{\tilde{j}}$ from the other prices can fit $p_{\tilde{j}}$ perfectly. Thus we must have

[24] If all prices are trivial, then we will achieve a contradiction, because that implies that the classifier does do at least as well as the edge-over-random guesser, since all classifiers achieve the same payoff.

$$c_{\tilde{j},y_{\tilde{j}}} < \mathbb{E}_{\hat{y}\sim\text{Unif}}[c_{\tilde{j},\hat{y}}].$$

So consider the hyperplane classifier which predicts $\tilde{y}$ for $\tilde{p}$, and y^* for $p \in P\backslash\{\tilde{p}\}$, i.e., depending on which side of h^* they are on (acknowledging that this may be a trivial classifier). Denote the resulting classifier by h. For this single-hyperplane classifier, we have

$$\sum_{p_j\in P} D^*(p_j)c_{j,h(p_j)} = D^*(p_{\tilde{j}})c_{\tilde{j},y_{\tilde{j}}} + \left(\sum_{q\in P\backslash\{p_{\tilde{j}}\}} D^*(q)\right)\sum_{p_k\in P\backslash\{p_{\tilde{j}}\}} \frac{D^*(p_k)}{\sum_{q\in P\backslash\{p_{\tilde{j}}\}} D^*(q)}c_{k,y^*} > U^0_{c^*},$$

where the inequality holds since the single-threshold classifier does strictly better on some non-trivial price, and as well on all other prices. This completes the proof. □

B. Proofs of the Main Results 8.

B.1. Convergence of τ_A

B.1.1. The $|A| = 2$ case

We replicate the proof in Schapire and Freund (2012) for reference. Define

$$F_t(p) = \sum_{k=1}^{t} \alpha_k h_k(p).$$

Following the same recursive process described in Schapire and Freund (2012), we have

$$d_{t+1}(p) = \frac{d_1(p)\exp\left(-y(\sigma,p)\sum_{k=1}^{t}\alpha_k h_k(p)\right)}{\prod_{k=1}^{t} Z_k} = \frac{d_1(p)\exp(-y(\sigma,p)F_t(p))}{\prod_{k=1}^{t} Z_k}. \tag{B.8}$$

Following Schapire and Freund (2012), we can show that

$$\mathbb{P}\left(H_t(p) \neq y(\sigma,p)\right) = \mathbf{E}\sum_p d_1(p)\mathbf{1}(H_t(p) \neq y(\sigma,p)) \leq \mathbf{E}\sum_p d_1(p)\exp(-y(\sigma,p)F_t(p)),$$

and

$$\mathbb{P}(H_t(p) \neq y(\sigma,p)) = \mathbf{E}\prod_{k=1}^{t} Z_k.$$

Note

$$Z_k = \sum_p d_k(p)\exp\left(-y(\sigma,p)\alpha_k h_k(p)\right).$$

The rest of the proof follows from Schapire and Freund (2012), which we copy here for later reference.

$$\begin{aligned} Z_t &= \sum_p d_t(p) \exp\big(-y(\sigma,p)\alpha_t h_t(p)\big) \\ &= \sum_{y(\sigma,p)h_t(p)=1} d_t(p)\exp(-\alpha_t) + \sum_{y(\sigma,p)h_t(p)=-1} d_t(p)\exp(-\alpha_t) \\ &= e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t \\ &= e^{-\alpha_t}\left(\frac{1}{2}+\gamma_t\right) + e^{\alpha_t}\left(\frac{1}{2}-\gamma_t\right) \\ &= \sqrt{1-4\gamma_t^2} \end{aligned}$$

where

$$\gamma_t = \frac{1}{2} - \epsilon_t.$$

By weak learnability, we know that γ_t is uniformly bounded away from 0: $\exists \gamma > 0$ such that

$$\gamma_t \geq \gamma \qquad \forall t \geq 1.$$

Recall that the maximum number of the elements in the support of σ is N. Thus,

$$d_{t+1}(p) = d_1(p)\prod_{k=1}^{t}\sqrt{1-4\gamma_t^2} \leq \frac{1}{N}\left(1-4\gamma^2\right)^{\frac{t}{2}} \leq \frac{1}{N}e^{-2\gamma^2 t}$$

where the right hand side converges to 0 at the exponential rate uniformly over p.

B.2. The $|A| > 2$ case

The specification of the algorithm can be found in Mukherjee and Schapire (2013). The proof provided below fills in some details to show that convergence holds in a self-contained way.

First, initialize $F_y^0(x_i) = 0$.

- From previous stage, take F_y^t.
- At stage t, find the $h \in \mathcal{H}$ solving:

$$\min_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}[h_t(x_i) = y_i]\left((e^{-\eta}-1)\sum_{\tilde{y}\neq y_i} e^{\eta(F_{\tilde{y}}^{t-1} - F_{y_i}^{t-1})}\right) + \mathbf{1}[h_t(x_i) \neq y_i](e^{\eta}-1)e^{\eta(F_{h_t(x_i)}^{t-1} - F_y^{t-1}(x_i))}.$$

- Define $F_y^t(x_i) = \sum_{s=1}^{t} \mathbf{1}[h_t(x_i) = y]$.

The final prediction is $H_t(x_i) = \arg\max_{\tilde{y}} \sum_{t=1}^{T} \mathbf{1}[h_t(x_i) = \tilde{y}]$.

The weak learnability condition says that the hypothesis class can outperform a random guesser that does better than some γ, where we allow for a potentially asymmetric cost of making different errors.

We now show convergence to the rational rule:

Step 1: Bounding The Mistakes: This step is as previous. We have

$$\sum_{i=1}^{m} \mathbf{1}[H_t(x_i) \neq y_i] \leq \sum_{i=1}^{m} \sum_{\tilde{y} \neq y_i} e^{\eta(F^t_{\tilde{y}}(x_i) - F^t_{y_i}(x_i))}.$$

Indeed, the exponential is positive, so this inequality holds when y_i is labelled correctly, and if the label is incorrect, then that means that some $\tilde{y}_i$ satisfies $F^t_{\tilde{y}_i}(x_i) > F^t_{y_i}(x_i)$. Since all exponential terms are positive, and furthermore the exponent is positive if x_i is labelled incorrectly, meaning the right hand side is greater than 1 if mislabeled.

Step 2: Recursive Formulation of the Loss We now show that the right hand side goes to 0 at an exponential rate. We define the loss function to be:

$$L_t(x_i) = \sum_{\tilde{y} \neq y_i} e^{\eta(F^t_{\tilde{y}}(x_i) - F^t_{y_i}(x_i))}, \tilde{L}_t = \frac{1}{m} \sum_{i=1}^{m} L_t(x_i).$$

We first express $\tilde{L}_{t+1}$ as a function of $\tilde{L}_t$. Note that $F^{t+1}_{\tilde{y}}(x_i) = F^t_{\tilde{y}}(x_i)$ for all $\tilde{y} \neq h_t(x_i)$, and $F^{t+1}_{\tilde{y}}(x_i) = F^t_{\tilde{y}}(x_i) + 1$ for $\tilde{y} = h_t(x_i)$. The loss from a given x_i changes depending on whether or not it is correctly classified. For any observation that is classified correctly at the $t+1$th stage, we multiply that observation's loss by a factor of $e^{-\eta}$. On the other hand, for any observation that is classified incorrectly as $\tilde{y}$, we *add* the following:

$$e^{\eta(F^t_{\tilde{y}}(x_i) - F^t_{y_i}(x_i))}(e^{\eta} - 1).$$

So:

$$\tilde{L}_{t+1} = \frac{1}{m} \left(\sum_{i : h_{t+1}(x_i) = y_i} e^{-\eta} L_t(x_i) + \sum_{i : h_{t+1}(x_i) \neq y_i} \left(L_t(x_i) + e^{\eta(F^t_{h_{t+1}(x_i)}(x_i) - F^t_{y_i}(x_i))}(e^{\eta} - 1) \right) \right).$$

Note that if we subtract $\tilde{L}_t$ from both sides, and substitute in for $L_t(x_i)$ above, we obtain:

$$\tilde{L}_{t+1} - \tilde{L}_t = \frac{1}{m} \left(\sum_{i : h_{t+1}(x_i) = y_i} (e^{-\eta} - 1) \sum_{\tilde{y} \neq y_i} e^{\eta(F^t_{\tilde{y}}(x_i) - F^t_{y_i}(x_i))} + \sum_{i : h_{t+1}(x_i) \neq y_i} e^{\eta(F^t_{h_{t+1}(x_i)}(x_i) - F^t_{y_i}(x_i))}(e^{\eta} - 1) \right).$$

Step 3: Weak Learnability By the above, h_{t+1} is chosen to solve:

$$\min_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}[h(x_i) = y_i] \left((e^{-\eta} - 1) \sum_{\tilde{y} \neq y_i} e^{\eta(F^t_{\tilde{y}}(x_i) - F^t_{y_i}(x_i))} \right) + \mathbf{1}[h(x_i) \neq y_i](e^{\eta} - 1) e^{\eta(F^t_{h(x_i)}(x_i) - F^t_{y}(x_i))}.$$

In fact, using the previous step, we see that this can equivalently be expressed as $\tilde{L}_{t+1} - \tilde{L}_t$. On the other hand, someone who is random guessing, but is correct with extra probability γ, will be correct with probability $\frac{1-\gamma}{k} + \gamma$, and guess an incorrect label $\tilde{y}$ with probability $\frac{1-\gamma}{k}$. Furthermore, the hypothesis class ensures a weakly lower error (as measured by this cost) than the random guessing. Hence this expression is bounded above by:

$$\frac{1}{m}\sum_{i=1}^{m}\left((\frac{1-\gamma}{k}+\gamma)(e^{-\eta}-1)L_t(x_i)+\frac{1-\gamma}{k}\sum_{\tilde{y}\neq y_i}(e^{\eta}-1)e^{\eta(F_{\tilde{y}}^t(x_i)-F_y(x_i))}\right)$$

Again substituting in for $L_t(x_i)$ and rearranging, we obtain:

$$\left((\frac{1-\gamma}{k}+\gamma)(e^{-\eta}-1)+\frac{1-\gamma}{k}(e^{\eta}-1)\right)\tilde{L}_t.$$

Putting this together, we have this is an upper bound of $\tilde{L}_{t+1}-\tilde{L}_t$, and therefore:

$$\tilde{L}_{t+1}\leq\left(1+\left((\frac{1-\gamma}{k}+\gamma)(e^{-\eta}-1)+\frac{1-\gamma}{k}(e^{\eta}-1)\right)\right)\tilde{L}_t.$$

Step 4: Specifying η We are done if we can ensure $\tilde{L}_t\to 0$ as $t\to\infty$, since Step 1 shows that this implies that the number of misclassifications approaches 0 as well. To complete the argument, we must specify an η which delivers the exponential convergence. However, first note that if $\eta=0$, the coefficient on $\tilde{L}_t$ in the previous inequality is 1, and the derivative with respect to η is $-\gamma$ at 0, so that this expression is less than 1, for some $\eta>0$. Setting $\eta=\log(1+\gamma)$, the above coefficient on $\tilde{L}_t$ reduces to:

$$\overbrace{1+\left((\frac{1-\gamma}{k}+\gamma)(\frac{1}{1+\gamma}-1)+\frac{1-\gamma}{k}\gamma\right)}^{z_k(\gamma)}.$$

Note that $z_k(\gamma)$ is bounded above by $\tilde{z}(\gamma)=e^{-\gamma^2/2}$. Indeed, this expression is decreasing in k, with $z_k(0)=1=\tilde{z}(0)$, and $z_2(\gamma)=1-\frac{\gamma^2}{2}<e^{-\gamma^2/2}=\tilde{z}(\gamma)$. Since $\tilde{L}_0=(k-1)$, we therefore have that:

$$\tilde{L}_t\leq(k-1)e^{-\gamma t^2/2},$$

as desired.

B.3. Convergence of $\tau_{\hat{A}}$

B.3.1. Preliminaries

Let $\pi(v:p)$ be the posterior distribution of v conditioned on p, and note that v is drawn from a finite set, then $\pi(v:p)$ is a multinomial distribution. Let $\hat{\pi}_t(v:p)$ be the sample average for $\pi(v:p)$. We know that the rate function of $\hat{\pi}_t(v:p)$ is the relative entropy of $\hat{\pi}_t$ with respect to π (Dembo and Zeitouni (1998))

$$I_\pi=\sum_v\hat{\pi}_t(v:p)\log\frac{\hat{\pi}_t(v:p)}{\pi(v:p)},$$

from which we derive λ in (6.7): $\forall\epsilon>0$, let $N_\epsilon(\pi)$ be the ϵ neighborhood of $\pi(v:p)$, and

$$\lambda=\inf_{\hat{\pi}_t\notin N_\epsilon(\pi)}I_\pi.$$

Note that

$$y^R(\sigma,p)\neq\hat{y}_t(p)$$

only if π and $\hat{\pi}_t$ prescribe different actions. Since $\hat{\pi}_t$ is a consistent estimator of π, the probability of two probability distributions prescribing two different actions vanishes. The large deviation property of $\hat{\pi}_t$ implies that $\hat{y}_t(p)$ satisfies (6.7), if $y^R(\sigma, p)$ is a strict best response.

By the concavity of the logarithmic function, I_π is minimized if π is a uniform distribution and

$$\inf_{\pi} I_\pi > 0.$$

If$|P| < \infty$ and$|A| < \infty$, we obtain the uniform version of (6.7) with respect to the true probability distribution. We state the result without proof for later reference.

Lemma 4. *Suppose that $\hat{y}_t(p)$ is a consistent*[25] *estimator of $y^R(\sigma, p)$ and satisfies* (6.7). *Then, $\exists \lambda > 0$ such that*

$$\limsup_{t\to\infty} -\frac{1}{t} \log \mathbb{P}\left(y^R(\sigma, p) \neq \hat{y}_t(p) \;\; \forall p \textit{ in the support of } \sigma \right) \leq \lambda. \tag{B.9}$$

We construct algorithm $\tau_{\hat{A}}$ by replacing $y(\sigma, p)$ by $\hat{y}_t(p)$ in τ_A constructed in the previous section. More precisely, let $f_t^y(p)$ be the empirical probability that $\hat{y}_t(p) = 1$ at the beginning of period t. Thus, $\hat{y}_t(p) = -1$ with probability $1 - f_t^y(p)$. Given $\{d_t(p), \hat{y}_t(p)\}_p$, h_t solves

$$\max_{h\in\mathcal{H}} \sum_p h(p) d_t(p) [1 \cdot f_t^y(p) - 1 \cdot (1 - f_t^y(p))]$$

and

$$\hat{\epsilon}_t = \sum_p d_t(p) \left[f_t^y(p) \mathbf{1}(h(p) = 1) + (1 - f_t^y(p)) \mathbf{1}(h(p) = -1) \right].$$

Using weak learnability, we can show that $\exists \rho > 0$ such that

$$\hat{\epsilon}_t \leq \frac{1}{2} - \rho.$$

Since $\hat{y}_t(p)$ has the full support over $\{-1, 1\}$ $\forall t \geq 1$,

$$\hat{\epsilon}_t > 0.$$

Given an algorithm τ_A with observed labels, we can therefore replace it with $\tau_{\hat{A}}$ which involves inferring the labels $y^R(\sigma, \cdot)$, setting them equal to $\hat{y}_t(\cdot)$, for all $t \geq 1$.

B.3.2. Main Proof

Under the assumption that $y^R(\sigma, p)$ is a strict best response,

$$\lim_{t\to\infty} \hat{y}_t(p) = y^R(\sigma, p)$$

almost surely. Since $\hat{y}_t(p)$ satisfies the uniform LDP, $\forall \epsilon > 0$, $\exists \rho(\epsilon, \sigma) > 0$ and $T(\epsilon, \sigma)$ such that

$$\mathbb{P}\left(\exists t \geq T(\epsilon, \sigma), \hat{y}_t(p) \neq y^R(\sigma, p) \right) \leq e^{-t\rho(\epsilon,\sigma)}.$$

[25] An estimator $\hat{y}_t(p)$ is consistent if $\hat{y}_t(p)$ converges to $y^R(\sigma, p)$ in probability as $t \to \infty$.

Since the support of σ contains a finite number of p, the empirical the multinomial probability distribution over θ.

Let $\hat{\pi}_t(\theta : p)$ be the empirical probability distribution over Θ following t rounds of observations. By the law of large numbers, $\hat{\pi}_t(\theta : p) \to \pi(\theta : p)$ computed via Bayes rule from the prior distribution over θ and σ. Write $\Theta = (\theta_1, \ldots, \theta_{|\Theta|})$. Given $\epsilon = (\epsilon, \ldots, \epsilon) \in \mathbb{R}^{|\Theta|}$, the rate function of the multinomial distribution is

$$\sum_{i=1}^{|\Theta|} \epsilon \log \frac{\epsilon}{p(\theta)}$$

where $p(\theta)$ is the probability that θ is realized. Since $\sum_\theta p(\theta) = 1$,

$$\sum_{i=1}^{|\Theta|} \epsilon \log \frac{\epsilon}{p(\theta)} \geq \prod_{i=1}^{|\Theta|} \epsilon \log \frac{\epsilon}{1/|\Theta|} = \prod_{i=1}^{|\Theta|} \epsilon \log \epsilon |\Theta| > 0.$$

Note that the right hand side is independent of σ, which is the rate function of the uniform distribution over Θ. Thus, we can choose $\rho(\epsilon) \leq \rho(\epsilon, \sigma)$ uniformly over σ, which is strictly increasing with respect to $\epsilon > 0$. We choose $T(\epsilon)$ independently of σ as well.

Define an event

$$\mathcal{L} = \left\{ \hat{y}_t(p) = y^R(\sigma, p) \qquad \forall t \geq T(\epsilon) \right\}$$

We know that

$$\mathbb{P}(\mathcal{L}) \geq 1 - e^{-t\rho(\epsilon)}.$$

Fix $t > T(\epsilon)$. We have

$$\begin{aligned}
& \mathbb{P}\left(\tau_{\hat{A}}(D_t)(p) \neq y^R(\sigma, p) \right) \\
= \;& \mathbb{P}\left(\tau_{\hat{A}}(D_t)(p) \neq y^R(\sigma, p) : \mathcal{L} \right) \mathbb{P}(\mathcal{L}) + \mathbb{P}\left(\tau_{\hat{A}}(D_t)(p) \neq y^R(\sigma, p) : \mathcal{L}^c \right) \mathbb{P}(\mathcal{L}^c) \\
\leq \;& \mathbb{P}\left(\tau_{\hat{A}}(D_t)(p) \neq y^R(\sigma, p) : \mathcal{L} \right) + \mathbb{P}(\mathcal{L}^c) \\
\leq \;& \mathbb{P}\left(\tau_{\hat{A}}(D_t)(p) \neq y^R(\sigma, p) : \mathcal{L} \right) + e^{-t\rho(\epsilon)}.
\end{aligned}$$

Following the same logic as in the proof of Proposition 8, we can show that $\exists \gamma(G) > 0$ such that

$$\hat{Z}_t \leq 1 - \gamma(G) \qquad \forall t \geq 1 \tag{B.10}$$

under $\tau_{\hat{A}}$.

Recall that

$$F_a(p) = \sum_{s=1}^{t} \alpha_s \mathbf{1}(h_s(p) = a).$$

Similarly, we define

$$\hat{F}_a(p) = \sum_{s=1}^{t} \hat{\alpha}_s \mathbf{1}(h_s(p) = a).$$

Following the same logic as in the proof of Proposition 8, we know that if $\tau_{\hat{A}}(D_t)(p) \neq y^R(p)$,

$$\hat{F}_{y^R(\sigma,p)}(p) + \sum_{a \neq y^R(\sigma,p)} \hat{F}_a(p) > 0.$$

Thus,

$$\begin{aligned}
\mathbf{1}(\tau_{\hat{A}}(D_t)(p) \neq y^R(\sigma,p)) &\leq \mathbf{1}\left(\hat{F}_{y^R(\sigma,p)}(p) + \sum_{a \neq y^R(\sigma,p)} \hat{F}_a(p)\right) \\
&\leq \exp\left(\hat{F}_{y^R(\sigma,p)}(p) + \sum_{a \neq y^R(\sigma,p)} \hat{F}_a(p)\right).
\end{aligned}$$

Conditioned on event $\mathcal{L}$,

$$\hat{y}_t(p) = y^R(\sigma,p) \qquad \forall t \geq T(\epsilon).$$

We can write for $t \geq T(\epsilon)$,

$$\begin{aligned}
d_{t+1}(p) &= \frac{\hat{d}_t(p)\exp(\alpha_t(\mathbf{1}(h_t(p) \neq \hat{y}_t(p)) - \mathbf{1}(h_t(p) = \hat{y}_t(p))))}{\hat{Z}_t} \\
&= \frac{\hat{d}_t(p)\exp(\alpha_t(\mathbf{1}(h_t(p) \neq y^R(\sigma,p)) - \mathbf{1}(h_t(p) = y^R(\sigma,p))))}{\hat{Z}_t} \\
&= \frac{d_{T(\epsilon)}(p)\exp(\sum_{s=T(\epsilon)}^{t} \alpha_s(\mathbf{1}(h_s(p) \neq y^R(\sigma,p)) - \mathbf{1}(h_s(p) = y^R(\sigma,p))))}{\prod_{s=T(\epsilon)}^{t} \hat{Z}_t}.
\end{aligned}$$

Thus,

$$\begin{aligned}
&\prod_{s=T(\epsilon)}^{t} \hat{Z}_t \\
= \;& \sum_p d_{T(\epsilon)}(p)\exp\left[\sum_{s=T(\epsilon)}^{t} \alpha_s(\mathbf{1}(h_s(p) \neq y^R(\sigma,p)) - \mathbf{1}(h_s(p) = y^R(\sigma,p)))\right] \\
\geq \;& \left(\min_{p \in \mathcal{P}(\sigma)} d_{T(\epsilon)}(p)\right) \sum_p \exp\left[\sum_{s=T(\epsilon)}^{t} \alpha_s(\mathbf{1}(h_s(p) \neq y^R(\sigma,p)) - \mathbf{1}(h_s(p) = y^R(\sigma,p)))\right].
\end{aligned}$$

Since $d_1(p)$ is the uniform distribution over $\mathcal{P}(\sigma)$,

$$\min_{p \in \mathcal{P}(\sigma)} d_{T(\epsilon)}(p) > 0.$$

We can write

$$\begin{aligned}
\prod_{s=1}^{t} \hat{Z}_t &= \prod_{s=T(\epsilon)}^{t} \hat{Z}_t \prod_{s=1}^{T(\epsilon)-1} \hat{Z}_t \\
&\geq \left(\min_{p\in\mathcal{P}(\sigma)} d_{T(\epsilon)}(p)\right) \sum_p \exp(\sum_{s=T(\epsilon)}^{t} \hat{\alpha}_s(\mathbf{1}(h_s(p) \neq y^R(\sigma,p)) - \mathbf{1}(h_s(p) = y^R(\sigma,p)))) \prod_{s=1}^{T(\epsilon)-1} \hat{Z}_t \\
&= \frac{\left(\min_{p\in\mathcal{P}(\sigma)} d_{T(\epsilon)}(p)\right) \prod_{s=1}^{T(\epsilon)-1} \hat{Z}_t}{\sum_p \exp\left[\sum_{s=1}^{T(\epsilon)-1} \hat{\alpha}_s(\mathbf{1}(h_s(p) \neq y^R(\sigma,p)) - \mathbf{1}(h_s(p) = y^R(\sigma,p)))\right]} \\
&\qquad \times \sum_p \exp\left[\sum_{s=1}^{t} \hat{\alpha}_s(\mathbf{1}(h_s(p) \neq y^R(\sigma,p)) - \mathbf{1}(h_s(p) = y^R(\sigma,p)))\right]
\end{aligned}$$

over $\mathcal{L}$. Define

$$M(\epsilon) = \frac{\left(\min_{p\in\mathcal{P}(\sigma)} d_{T(\epsilon)}(p)\right) \prod_{s=1}^{T(\epsilon)-1} \hat{Z}_t}{\sum_p \exp(\sum_{s=1}^{T(\epsilon)-1} \hat{\alpha}_s(\mathbf{1}(h_s(p) \neq y^R(\sigma,p)) - \mathbf{1}(h_s(p) = y^R(\sigma,p))))}$$

which is bounded away from 0.

Recall that

$$\begin{aligned}
& \mathbb{P}(\tau_{\hat{A}}(D_t)(p) \neq y^R(\sigma,p)) \\
\leq\ & \sum_p d_1(p) \exp(\sum_{s=1}^{t} \hat{\alpha}_s(\mathbf{1}(h_s(p) \neq y^R(\sigma,p)) - \mathbf{1}(h_s(p) = y^R(\sigma,p)))) \\
\leq\ & \frac{\prod_{s=1}^{t} \hat{Z}_t}{M(\epsilon)} \leq \frac{(1-\gamma(G))^t}{M(\epsilon)} \leq \frac{e^{-t\gamma(G)}}{M(\epsilon)}.
\end{aligned}$$

Combining the probabilities over $\mathcal{L}$ and $\mathcal{L}^c$, we have that $\forall\epsilon$, $\forall\sigma \in \Sigma^G \subset \Sigma$, $\exists T(\epsilon)$, $\rho(\epsilon)$ and $\gamma(G)$ such that

$$\mathbb{P}\left(\exists t \geq T(\epsilon),\ \tau_{\hat{A}}(D_t)(p) \neq y^R(\sigma,p)\right) \leq \frac{e^{-t\gamma(G)}}{M(\epsilon)} + e^{-t\rho(\epsilon)}.$$

We can choose $T > T(\epsilon)$ and $\overline{\rho}$ such that $\forall t \geq T$,

$$\frac{e^{-t\gamma(G)}}{M(\epsilon)} + e^{-t\rho(\epsilon)} \leq e^{-\overline{\rho}t}$$

which proves the proposition.

C. Accommodating Richer $\mathcal{P}$ Spaces (Section 6.3.2)

C.1. Approaches

Approach One: Discretization We describe how to revise $\tau_{\hat{A}}$ accordingly to discretize the observation space. Instead of processing individual actions, we let $\tau_{\hat{A}}$ process a group of actions at a time, treating "close" actions as the same group. In principle, we want to partition $\mathcal{P}$ into a set of half-open rectangles intervals with size λ. More precisely, given some arbitrary λ, we can partition each dimension of a rectangle containing $\mathcal{P}$ into the collection of half open intervals of size $\lambda > 0$ with a possible exception of the last interval:

$$P_0^j = [\underline{p}, \underline{p} + \lambda), \ldots, P^j_{K_j^\lambda} = [\underline{p} + (K_j^\lambda - 1)\lambda), \overline{p}]$$

where K_j^λ is the number of elements in the partition and $j \in \{1, \ldots, n\}$ is a partiular dimension.

For each element in the partition, the algorithm receives an ordinal information about the average outcome from the decision, if it contains a sender action in the support of σ:

$$\hat{y}_t^\lambda(k) = a \text{ if } a = \arg\max \sum_{p \in P_k} v(\theta, p, a)$$

where p in the support of σ and P_k is the product of partition elements. Let $\tau_{\hat{A}}^\lambda$ be the algorithm obtained by replacing $\hat{y}_t(p)$ in $\tau_{\hat{A}}$ by $\hat{y}_t^\lambda(k)$. Note that as $\lambda \to 0$, the size of the individual elements in the partition shrinks and $\tau_{\hat{A}}^\lambda$ converges to $\tau_{\hat{A}}$ for a fixed σ.

Compared to τ_A and $\tau_{\hat{A}}$, $\tau_{\hat{A}}^\lambda$ takes only coarse information for two important reasons. First, the algorithm cannot differentiate two ps which are very close. This features makes the algorithm robust against strategic manipulation of the sender to slow down the speed of learning. Second, the algorithm cannot detect the precise consequence of its decision, but only the ordinal information of the past decision, aggregated over time. The second feature allows the algorithm to operate with very little information about the details of the parameters of the underlying game.

Approach Two: Smoothing Discretizing the action space as above is one way of ensuring that there are only a finite number of sender actions to worry about in the long run, and given a sufficiently fine discretization, any distinct p is distinguished by the algorithm. However, in principle, close sender actions may still be quite far in terms of payoffs, and only be distinguished in the long run. That is, there is no guarnatee that for a fixed horizon, that the algorithm is not grouping too many p possibilities. The issue is that the discretization approach uses no information about the receiver's payoff function. Our other alternative describes more explicitly how *close* to rationality the receiver can achive, given some fixed discretization scheme.

The idea is the following: We add a small amount of noise to each observed p, with the amount of noise tending to 0 as the sample size grows large. Doing so allows us to show that the receiver perceives the sender's strategy to have the property that $\mathbb{E}_\theta[u(a, \theta, p(a) : p]$ is uniformly equicontinuous (as functions of p). As a result, if the receiver only seeks to use a strategy that is $\varepsilon-$optimal against σ, uniform equicontinuity implies that their best reply can essentially be collapsed within intervals.

It will additionally be important that the algorithm does not seek to make predictions at p values where the corresponding density would be estimated to be small. Hence a second step will be to determine whether a p realization occur in a region with sufficiently large probability, where the "sufficient" amount will also tend to 0 as the amount of data grows large.

Formally, suppose the algorithm observes data $((p_1, y), \ldots, (p_n, y))$. Let $z_{\eta,i}$ be an independent random vector in the unit ball around 0 distributed according to the PDF:

$$\phi_\eta(z) = \frac{1}{K} \exp\left(-\frac{1}{1 - |z/\eta|^2}\right) \frac{1}{\eta^{|A|-1}},$$

where K is a constant which ensures ϕ_η integrates to 1. Our first augmentation is the following:

- Replace the observed $p_1, \ldots, p_n$ with $\hat{p}_1, \ldots, \hat{p}_n$, where $\hat{p}_i = p_i + z_{\eta,i}$, with $z_{\eta,i}$ distributed according to the above.

Second, it turns out that the above smoothing operation only works if the density is sufficiently large. Otherwise, the smoothing noise has too much power.

- For any $\tilde{p} = (\tilde{p}_a)_{a \in A \setminus a_0}$ drawn, estimate the event that $\tilde{\sigma}_\eta(\tilde{p}) < \gamma$ by fixing some δ small and determining whether menu(s) p with $\max_{a \in A \setminus \{a_0\}} \tilde{p}_a - p_a < \delta$ occurs with frequency at least $(2\delta)^{|A|-1}\gamma$. Recommend action a_0 for any such p.

As $\delta \to 0$, the condition holds if the density is at least γ. Together with the previous, we can show that if the receiver instead observes noisy sender actions, the perceived sender's strategy is sufficiently well-behaved to maintain the appropriate convergence for the algorithm.

Proposition 9. *Suppose the sender is restricted to choosing distributions which are either discrete or continuous with bounded density. Consider an algorithm which can ensure that an ε-rational label is PAC-learnable, for any arbitrary $\varepsilon > 0$ given a finite number of possible sender actions. Then there exists a smoothing operation which maintains PAC-learnability of ε-rationality, for every $\varepsilon > 0$.*

The idea of the proposition is to use the smoothing operation to show that the algorithm perceives that the sender uses a σ such that $\mathbb{E}[u(a, \theta, p(a)) : p]$ is uniformly equicontinuous. Given that we seek ε-optimality, uniform equicontinuity allows us to essentially discretize the menu space, transforming the environment into a much simpler one.

There are two important properties of the transformation which allows us to ensure this works. The first is that, defining $\tilde{\sigma}_\eta(\cdot : \theta)$ to be the perceived p distribution of $p_i + z_i$, we have:

$$D^\alpha \tilde{\sigma}_\eta(p : \theta) = \int_P D^\alpha \phi_\eta(p - \tilde{p}) \sigma(\tilde{p} : \theta) d\tilde{p},$$

so that $\tilde{\sigma}_\eta$ inherits the smoothness properties of ϕ_η. The second is that, on any compact subset of P, we have $\sigma_\eta(\cdot : \theta) \to \sigma(\cdot : \theta)$ uniformly. Now, in order to obtain uniform continuity as $\eta \to 0$, it will be important that we can simultaneously ensure that the sender's strategy does not involve dramatic movements in the conditional probability. For instance, suppose the sender were to use the following strategy:

$$\sigma(p : G) = p(\sin\left(\frac{1}{p}\right) + 1), \sigma(p : B) = p(\sin\left(\frac{1}{p} - \pi\right) + 1),$$

defined on an interval $[0, \overline{p}]$ such that both densities integrate to 1. Then $\mathbb{P}[\theta = G : p] = 1$ if $p = \frac{1}{(2k+1/2)\pi}$ for some $k \in \mathbb{N}$, and 0 if $p = \frac{1}{(2k+1/2)\pi}$, for some $k \in \mathbb{N}$. As $k \to \infty$ (so that $p \to 0$), this oscillates infinitely often.

We handle the problem this example poses by only making non-degenerate predictions if the probability of using such sender actions is sufficiently high. That is, we "ignore" p realizations which only occur with low probability according to an estimated density.[26] Seeking to estimate the probability that all sender actions are within δ of p in order to estimate the density is just one way of doing this step; for instance, one could estimate the CDF $\tilde{\sigma}_\eta(p)$, and

[26] One may wonder why this trick works; for instance, we do not obtain the result when $\sigma(p : G) = \sin\left(\frac{1}{p}\right) + 1, \sigma(p : B) = \sin\left(\frac{1}{p} - \pi\right) + 1$. However, unlike the previous example, these will fail the continuity requirement on the sender's strategy space, which is needed in the proof.

use the estimated density to determine whether the observations should be thrown away. Ultimately, however, given the compact $\mathcal{P}$, we can minimize the probability that this is done by using sufficiently low thresholds. As a result, it has a vanishing impact on PAC-learnability, as well as the sender's expected profit.

C.2. Proof of Proposition 9

The proof of the theorem proceeds in the following steps:

- Step 1: Show that the expected value conditional on price, in the image of the sender's possible strategies after applying the augmentation, is uniformly equicontinuous.
- Step 2: Show that the same label is applied to $\mathbb{E}[v_\theta \mid p + z_{i,\eta}, \sigma, \phi_\eta]$ as would be applied to $\mathbb{E}[v_\theta \mid p, \sigma]$, with high probability.
- Step 3: Verify that the change in recommendation due to discarding "low density prices" occurs with vanishing probability.

Putting these together shows that the change in the expectation can be made arbitrarily small, as can the probability that small density observations are drawn. The condition that σ is either discrete or continuous is stronger than necessary; what is necessary is continuity of the conditional expectation as a function of price, which can be satisfied if the discrete portions and continuous portions are separated, for instance. However, the proposition highlights that we need not restrict the sender's strategy space at all in order for our algorithm to converge.

The Theorem implies that if the sender were to use an *arbitrary* strategy σ, the receiver could instead focus on finding a rational response to $\tilde{\sigma}_\eta$. Doing so would still lead to PAC learnability of the approximately optimal response to σ. On the other hand, we can show that the optimal response to $\tilde{\sigma}_\eta$ is PAC learnable (unlike, potentially, the optimal response to σ), and doing the change leads to a negligible impact on the sender's surplus.

Before presenting the proof, we argue that uniform equicontinuity implies weak learnability. Suppose that $\mathbb{E}[v \mid \sigma, p] - p$ is uniformly equicontinuous (which holds if $\mathbb{E}[v \mid \sigma, p]$ is uniformly equicontinuous). By uniform equicontinuity, we have there exists some δ such that whenever $\left|p - p'\right| < \delta$, we have that

$$\left|\mathbb{E}[v \mid \sigma, p] - \mathbb{E}[v \mid \sigma, p']\right| < 2\varepsilon,$$

for any σ. Suppose we have some price p such that $\mathbb{E}[v \mid \sigma, p] - p > \varepsilon$. Then if $\mathbb{E}[v \mid \sigma, p'] - p' < -\varepsilon$, it follows that $\left|p - p'\right| > \delta$. It follows that there can only be at most $\frac{v_H - v_L}{\delta}$ prices such that $y(\sigma, p) = -y(\sigma, p')$, where p and p' are adjacent (ignoring all prices where $\left|\mathbb{E}[v \mid \sigma, p] - p\right| < \varepsilon$, as the classification decision is irrelevant there).

C.2.1. Step One

We first show that $\mathbb{E}[v_\theta \mid \tilde{\sigma}_\eta, p]$ is Lipschitz in p uniformly of $\tilde{\sigma}_\eta$, noting that we are restricting to prices where $\tilde{\sigma}_\eta(p) > \gamma$. Note that:

$$\tilde{\sigma}'_\eta(p \mid \theta) = \int \phi'_\eta(p - \tilde{p})\sigma(\tilde{p} \mid \theta)d\tilde{p} \leq \max \phi'_\eta := \overline{\phi'}.$$

Furthermore, we have:

$$\frac{d}{dp}\mathbb{P}_{\tilde{\sigma}_\eta}[\theta \mid p] = \frac{\tilde{\sigma}'_\eta(p \mid \theta)\mathbb{P}[\theta]}{\sum_{\tilde{\theta}} \tilde{\sigma}_\eta(p \mid \tilde{\theta})\mathbb{P}[\tilde{\theta}]} - \frac{\tilde{\sigma}_\eta(p \mid \theta)\mathbb{P}[\theta](\sum_{\tilde{\theta}} \sigma'_\eta(p \mid \tilde{\theta})\mathbb{P}[\tilde{\theta}])}{(\sum_{\tilde{\theta}} \tilde{\sigma}_\eta(p \mid \tilde{\theta})\mathbb{P}[\tilde{\theta}])^2},$$

so:

$$\left|\frac{d}{dp}\mathbb{P}_{\tilde{\sigma}_\eta}[\theta \mid p]\right| \leq \overline{\phi'}\mathbb{P}[\theta] \cdot \left(\frac{1}{\sum_{\tilde{\theta}} \tilde{\sigma}_\eta(p \mid \tilde{\theta})\mathbb{P}[\tilde{\theta}]}\right) + \overline{\phi'}\left(\frac{\tilde{\sigma}_\eta(p \mid \theta)\mathbb{P}[\theta]}{(\sum_{\tilde{\theta}} \tilde{\sigma}_\eta(p \mid \tilde{\theta})\mathbb{P}[\tilde{\theta}])^2}\right) \leq \overline{\phi'}\mathbb{P}[\theta]\left(\frac{1}{\gamma} + \frac{M(\eta)}{\gamma^2}\right),$$

where $M(\eta)$ is a bound on $\tilde{\sigma}_\eta(p \mid \theta)\mathbb{P}[\theta]$, which exists since σ and ϕ_η have bounded densities. Hence we see that for all $p \neq p^*$, the conditional probability is uniformly bounded in p, and is hence Lipschitz continuous. Importantly, the bound only depends on η and γ (and $\mathbb{P}[\theta]$), and is therefore uniform over all strategies in the image of the augmentation. Hence we can ensure that Lipschitz continuity is mainted for all prices in the support of $\tilde{\sigma}_\eta$.

In fact, recall that the Lipschitz constant is equal to the L^∞ norm of the derivative. Hence Lipschitz continuity depends only on γ, $M(\eta)$ and $\overline{\phi'_\eta}$, meaning that the Lipschitz constant holds uniformly over the image of the distributions emerging under the algorithm. It follows that the image is uniformly equicontinuous.

C.2.2. Step Two

Note that since $\mathbb{E}[v_\theta \mid \sigma, p]$ is continuous on $S = \cup_\theta \ \text{Supp} \ \sigma(\cdot \mid \theta)$, $\mathbb{E}[v_\theta \mid \sigma, p]$ is uniformly continuous on any compact $K \subset S$. Define:

$$K_\gamma = \{p : \sum_\theta \sigma(p \mid \theta)\mathbb{P}[\theta] \geq \gamma\}.$$

Using that mollifiers converge uniformly on compact sets, we have that $\tilde{\sigma}_\eta \to \sigma$ uniformly on K_γ. We therefore have that, for any $\tilde{\varepsilon}$, we can find some $\overline{\eta}$ such that if $\eta < \overline{\eta}$ and $p \in K_\gamma$, then $\left|\tilde{\sigma}_\eta(p \mid \theta) - \sigma(p \mid \theta)\right| < \tilde{\varepsilon}$ for all θ, and $\left|\sum_\theta \tilde{\sigma}_\eta(p \mid \theta)\mathbb{P}[\theta] - \sum_\theta \sigma(p \mid \theta)\mathbb{P}[\theta]\right| < \tilde{\varepsilon}$.

Furthermore, since σ is uniformly continuous on K_γ, we have:

$$\left|\sigma(p \mid \theta) - \tilde{\sigma}(p' \mid \theta)\right| = \left|\int \phi_\eta(p' - \tilde{p})(\sigma(p \mid \theta) - \sigma(\tilde{p} \mid \theta))d\tilde{p}\right| \leq \tilde{\varepsilon},$$

using the uniform continuity of σ on K_γ.

So for any $p \in K_\gamma$, and η sufficiently small, we have (letting $\overline{v} = \max_\theta v_\theta$):

$$\begin{aligned}
\left|\mathbb{E}[v_\theta \mid \sigma, p] - \mathbb{E}[v_\theta \mid \tilde{\sigma}_\eta, p']\right| &= \left| \frac{\sum_\theta v_\theta \sigma(p \mid \theta)\mathbb{P}[\theta] \sum_{\tilde{\theta}} \tilde{\sigma}_\eta(p' \mid \tilde{\theta})\mathbb{P}[\tilde{\theta}] - \sum_\theta v_\theta \tilde{\sigma}_\eta(p' \mid \theta)\mathbb{P}[\theta] \sum_{\tilde{\theta}} \sigma(p \mid \tilde{\theta})\mathbb{P}[\tilde{\theta}]}{\left(\sum_\theta \sigma(p \mid \theta)\mathbb{P}[\theta]\right)\left(\sum_\theta \tilde{\sigma}_\eta(p' \mid \theta)\mathbb{P}[\theta]\right)} \right| \\
&\leq \frac{1}{\sigma(p) \cdot (\gamma - \tilde{\varepsilon})} \Bigg| \sum_\theta v_\theta(\sigma(p \mid \theta) - \tilde{\sigma}_\eta(p' \mid \theta))\mathbb{P}[\theta] \sum_{\tilde{\theta}} \sigma(p \mid \tilde{\theta})\mathbb{P}[\tilde{\theta}] \\
&+ \sum_\theta v_\theta \sigma(p \mid \theta)\mathbb{P}[\theta] \sum_{\tilde{\theta}} (\tilde{\sigma}_\eta(p' \mid \tilde{\theta}) - \sigma(p \mid \tilde{\theta}))\mathbb{P}[\tilde{\theta}] \Bigg| \\
&\leq \frac{1}{\sigma(p) \cdot (\gamma - \tilde{\varepsilon})} \Bigg(\overbrace{\left| \sum_\theta v_\theta(\sigma(p \mid \theta) - \tilde{\sigma}_\eta(p' \mid \theta)) \sum_{\tilde{\theta}} \sigma(p \mid \tilde{\theta})\mathbb{P}[\tilde{\theta}] \right|}^{\leq \overline{v}\tilde{\varepsilon}\sigma(p)} \\
&+ \overbrace{\left| \sum_\theta v_\theta \sigma(p \mid \theta)\mathbb{P}[\theta] \sum_{\tilde{\theta}} (\tilde{\sigma}_\eta(p' \mid \tilde{\theta}) - \sigma(p \mid \tilde{\theta}))\mathbb{P}[\tilde{\theta}] \right|}^{\leq \overline{v} \cdot \tilde{\varepsilon} \cdot \sigma(p)} \Bigg) \\
&\leq \frac{2\overline{v}\tilde{\varepsilon}}{\gamma - \tilde{\varepsilon}}.
\end{aligned}$$

The first inequality follows from adding and subtracting $\sum_\theta v_\theta \sigma(p \mid \theta)\mathbb{P}[\theta] \sum_{\tilde{\theta}} \sigma(p \mid \tilde{\theta})\mathbb{P}[\tilde{\theta}]$ to the numerator inside the absolute value (as well as the lower bound on $\tilde{\sigma}_\eta(p)$), and the second inequality is from the triangle inequality, and the overbraced expression follows from $v_\theta \leq \overline{v}$ and uniform convergence of $\tilde{\sigma}_\eta$ to σ.

So for any fixed γ, we can find some some η such that whenever $\eta < \overline{\eta}$, we can ensure that on K_γ, $\left|\mathbb{E}[v_\theta \mid \tilde{\sigma}_\eta, p] - \mathbb{E}[v_\theta \mid \sigma, p]\right| < \varepsilon^*$, by choosing $\tilde{\varepsilon}$ sufficiently small so that $\frac{2\tilde{\varepsilon}}{\gamma(\gamma - \tilde{\varepsilon})} < \varepsilon^*$. It follows that if the receiver's classifier converges to a rule that is ε-optimal under $\tilde{\sigma}_\eta$, it converges to a rule that is $\varepsilon + \varepsilon^*$ optimal under σ. The probability that this fails to occur is simply the probability that the price is outside of K_γ, which can be made arbitrarily small by taking $\gamma \to 0$, since we can approximate the support of σ arbitrarily well.

C.2.3. Step Three

Note that, for an arbitrary continuous distribution f, if $p \sim f$ we have (for any compact K):

$$\mathbb{P}_f[L_\gamma] = \int_K \mathbf{1}[p : f(p) \leq \gamma] f(p) dp \leq \int_K \mathbf{1}[p : f(p) \leq \gamma] \gamma dp \leq \mu(K) \cdot \gamma,$$

where μ is Lebesgue measure. It follows that the probability that $p \in L_\gamma$, is small if γ is small, and furthermore that this probability can be made small uniformly, using only γ.

As shown by the claim above, by taking η small, we can ensure that the difference in the conditional expected value is small with high probability. By taking γ small, we ensure that the probability of a different outcome due to smoothing goes to 0, implying the result.

D. Proofs for Examples

Proof of Proposition 1. The ideas in this proof are largely borrowed from Rubinstein (1993), accommodating two additional features of our enviroment: (a) need to infer the strategy from observed data and (b) the generalized setting, but we provide the proof for completeness (while also highlighting which general properties of the utility functions drive the result). We construct a strategy σ^* for the sender that generates higher payoff than the equilibrium strategy σ^R, thus deriving the contradiction that σ^R is a best response to τ in the long run. More precisely, define $(p^*_\theta, a(\theta))$ to be the sender payoff-maximizing strategy. We show that the sender can induce the receiver to choose $a(\theta) \neq y_R(p^*_\theta)$.

Fix $\epsilon > 0$ small, and suppose $\Theta = \{\theta_L, \theta_H\}$. First suppose $v(\theta_L, p^*_L, a_1) = v(\theta_L, p^*_L, a_0)$. Let $\tilde{p} \in (p_L, p_H)$ satisfies $v(\theta_L, \tilde{p}, 1) < v(\theta_L, \tilde{p}, 0)$. (If p is multidimensional, we can take $\tilde{p}$ to be on the line segment connecting p^*_L and p^*_H) Set $\eta = v(\theta_L, \tilde{p}, 0) - v(\theta_L, \tilde{p}, 1) > 0$. We then choose $\epsilon, \epsilon_H, \epsilon_L > 0$ to satisfy

$$\pi(H)\epsilon_H < \pi(L)\epsilon_L, \tag{D.11}$$

and such that

$$\frac{\epsilon_L}{\epsilon_L + \eta} < \epsilon < \frac{\pi(L)\epsilon_L - \pi(H)\epsilon_H}{\pi(L)\epsilon_L}. \tag{D.12}$$

Under the increasing differences assumption, we can find $p_i(\epsilon_i)$ such that

$$\epsilon_i = v(\theta_i, p_i(\epsilon_i), 1) - v(\theta_i, p_i(\epsilon_i), 0).$$

Consider the following randomized pricing rule σ^* of the sender: in state H, $\tilde{p}_H(\epsilon_H)$ is chosen with probability 1. In state L, $p_L(\epsilon_L)$ is chosen with probability $1 - \epsilon$ and $\tilde{p}$ with probability ϵ.

Under this strategy, the optimal response following $\tilde{p}$ is 0, and this does not vanish as all other parameters tend to 0. However, the ex-post optimal decisions are 1 for both $\tilde{p}_L(\epsilon_L)$ and $\tilde{p}_H(\epsilon_H)$. Nevertheless, (D.12) implies first, the decisionmaker prefers to choose $a = 1$ if and only if $\tilde{p}_L(\epsilon_L)$ than choose $a = 1$ if and only if $\tilde{p}_H(\epsilon_H)$; and second, that the loss from choosing $a = 1$ following $\tilde{p}$ is larger than the loss from choosing $a = 0$ at $\tilde{p}_L(\epsilon_L)$. Putting this together, and taking $\epsilon, \epsilon_L, \epsilon_H \to 0$ shows this policy approximates the sender's optimum, as desired.

The case of $v(\theta_L, p^*_L, a_1) > v(\theta_L, p^*_L, a_0)$ is even more straightforward, since in this case the gain from choosing a_1 is non-vanishing, meaning that we can set $\varepsilon_L = 0$.

The verification that the optimal rule converges to this threshold when emerging from data is straightforward; any recursive learning algorithm generates $\{\phi_t\}$ which converges to $\phi \in \left(v_L - \epsilon_L, \frac{v_H + v_L}{2}\right)$ to emulate the best response of type 1 buyer against σ. Thus, the long run average payoff against such algorithm should be bounded from below by $\mathcal{U}^*_p - \epsilon$. □

Proof of Proosition 2. First, we describe an environment where, when the buyer has access to two-threshold rules, the seller can devise a strategy which delivers higher payoff than the rational benchmark, even though the rational benchmark is a two theshold rule. In other words, we show how the argument in Proposition 1 can deliver the same conclusion when the buyer can use double threshold rules, albeit under a modified environment. Specifically, let $\Theta = \{\theta_0, \theta_L, \theta_H\}$; payoffs are exactly as above when $\theta \in \{\theta_L, \theta_H\}$. When $\theta = \theta_0$, however, we have $v(\theta_0, p, 0) >> v(\theta_0, p, 1)$ with $\arg\max_p v(\theta_0, p, 0) = 0$, while $u(\theta_0, p, 0) > u(\theta_0, p, 1)$ for all possible prices (e.g., high production costs and low quality). We also take $\pi(0)$ (the prior probability of state θ_0) small but fixed. In this case, the rational benchmark is the same as in the case of Proposition 1, with the addition of the seller charging a price of 0 and the buyer not purchasing in state θ_0.

For this environment, the argument from Proposition 1 carries over unchanged, except where the seller always charges 0 in state θ_0. Specifically, using the same pricing strategy in states θ_L and θ_H as outlined above requires the buyer having access to a three-threshold strategy; as $\epsilon, \epsilon_L, \epsilon_H \to 0$, the buyer approaches indifference across each action, whereas the benefit of choosing 0 in state θ_0 does not vanish. Therefore, now the argument implies that this strategy induces the buyer uses a two threshold rule, whereby they choose to not purchase at prices $p \in \{0, p_H(\epsilon_H)\}$

and to purchase at price $p = p_L(\epsilon_L)$.

To complete the proof of the proposition, we now uses these two environments—i.e., the one described in the previous paragraphs as well as the one described in Proposition 1—to show that if the buyer has access to n thresholds, the seller can profitably deviate by using a strategy such that $y^R(\sigma, p)$ requires $n+1$ thresholds; the environment under consideration will be as described in the first part of this proof if n is even, and as described in the proof of Proposition 1 if n is odd. Below we describe the proof for n odd, noting it is an identical argument when n is even (using the slighlty different environment).

Choose ϵ, ϵ_L and ϵ_H as in the above proposition. Let $\{\tilde{p}_k\}_{k=1,\ldots,n}$ be such that $p_L < \tilde{p}_1 < \cdots < p_H$ satisfy $v(\theta_L, \tilde{p}_k, 1) < v(\theta_L, \tilde{p}_k, 1)$ for k odd and $v(\theta_H, \tilde{p}_k, 1) > v(\theta_H, \tilde{p}_k, 1)$ for k even. Let η_k be the absolute value of the differences of these inequalities; choose ϵ_L, ϵ_H and ϵ, ϵ' such that (D.11) is satisfied, as well as:

$$\frac{\epsilon_L}{\epsilon_L + 2\eta_k/(n+1)} < \epsilon < \frac{\pi(L)\epsilon_L - \pi(H)\epsilon_H(1-\epsilon')}{\pi(L)\epsilon_L}, \tag{D.13}$$

and (assuming $n > 1$)

$$\frac{\epsilon_H}{\epsilon_H + 2\eta_k/(n-1)} < \epsilon' \tag{D.14}$$

Again choosing $p_i(\epsilon_i)$ in the same way as in the proof of Proposition 1, we consider the following pricing rule:

- In state H, $\tilde{p}_H(\epsilon_H)$ is chosen with probability $1-\epsilon'$; with complementary probability, the price is $\tilde{p}_k$ for k even, each with equal probability.
- In state L, $p_L(\epsilon_L)$ is chosen with probability $1-\epsilon$; with complementary probability, the price is $\tilde{p}_k$ for k odd, each with equal probability.

As before, the payoff gain from choosing after $\tilde{p}_i$ does not vanish as other parameters tend to 0, for all i. On the other hand, the inequlaities (D.11), (D.13) and (D.14) imply: First, the buyer does worse by erring on $\tilde{p}_L(\epsilon_L)$ than $\tilde{p}_H(\epsilon_H)$; and second, the loss from erring on any $\tilde{p}_k$ is larger than the loss from erring at either $\tilde{p}_L(\epsilon_L)$ or $\tilde{p}_H(\epsilon_H)$. Putting these observations together, we have the optimal decision rule is always an n-threshold rule that errs at $p_H(\epsilon_H)$, where the buyer chooses action 0. Again taking $\epsilon, \epsilon_L, \epsilon_H, \epsilon' \to 0$ approximates the seller's optimum, as desired. □

Proposition 3 follows immediately from the following result, which we call Proposition 3'.

Proposition 3' *Suppose $u(\theta, p, 1) - u(\theta, p, 0)$ is constant in θ and weakly concave in p. Suppose further that $u(\theta, p^*, 1) > u(\theta, p^*, 0)$. Then there exists a single threshold classifier which the algorithm could commit to using which ensures the strategic player chooses p^* with probability 1.*

Proof of Proposition 3'. Concave differences implies that the set

$$K = \{p : u(\theta, p, 1) \geq u(\theta, p, 0)\}$$

is a convex set; if $u(\theta, p_i, 1) - u(\theta, p_i, 0) \geq 0$ for $i = 1, 2$, then the same conclusion holds for $\alpha p_1 + (1-\alpha)p_2$ for all $\alpha \in [0, 1]$. Therefore, given any p^* on the boundary, the supporting hyperplane theorem implies that we can find a linear hyperplane (λ, ω) tangent to this set at p^*.

Suppose the algorithm designer prescribes that the receiver choose $a = 1$ at any menu p such that $\lambda \cdot p \leq \omega$ and $a = 0$ otherwise. Note that having the receiver choose $a = 1$ therefore requires choosing p where the sender would rather the receiver choose action $a = 0$, by definition of K. Therefore, the strategic player cannot do any better than choosing $\sigma(p \mid \theta)$ which is a point mass at p^*. □

Proof of Lemma 1. It suffices to show that if $p > v_L$ and $\mathbf{E}(v|p) - p \geq 0$, then the expected profit from p is strictly less than $\pi_L v_L$. We write the proof in Rubinstein (1993) for the later reference. For any price p satisfying

$$\mathbb{P}(H|p)v_H + \mathbb{P}(L|p)v_L \geq p,$$

the revenue cannot exceed

$$\mathbb{P}(H|p)v_H + \mathbb{P}(L|p)v_L$$

but the cost is

$$\mathbb{P}(H|p)(1-r)c_2 + \mathbb{P}(H|p)rc_1.$$

Thus, the sender's expected payoff is at most

$$\mathbb{P}(L|p)v_L + \mathbb{P}(H|p)((1-r)(v_H - c_2) + r(v_H - c_1))$$

Because of the lemon's problem,

$$(1-r)(v_H - c_2) + r(v_H - c_1) < 0$$

and

$$\mathbb{P}(H|p) > 0$$

to satisfy

$$\mathbb{P}(H|p)v_H + \mathbb{P}(L|p)v_L \geq p > v_L.$$

Integrating over p, we conclude that the ex ante profit is strictly less than $\pi_L v_L$.

□

Proof of Lemma 2. Let (x, q) be some offer such that the agent is indifferent between accepting and rejecting, so that:

$$q - \mathbb{E}[\theta : (x, q)]x = 0$$

The principal's expected payoff is found by taking the expectation of $u(\theta, (x, q), a)$ over all realizations of x, q. By the law of iterated expectations, this occurs if and only if the principal's payoff is maximized following *each* realization of (x, q). We claim the principal is *not* indifferent between actions following any such (x, q). Indeed, letting $\mathbb{E}[\theta : (x, q)] = r$, indifference implies:

$$(1-r)f(I - rx) + rf(I - L + (1-r)x) = (1-r)f(I) + rf(I - L).$$

Note that equality holds if $f(y) = y$. This implies that both lotteries, whether or not the principal accepts, have the same expected values. However, if f is concave, then since $I > I - rx > I - L + (1-r)x > I - L$, it must be that the left hand side is strictly greater than the right hand side.

It follows that if indifference holds, the principal strictly prefers the agent accept the offer by slightly reducing x. □

References

Abreu, D., and A. Rubinstein (1988): "The Structure of Nash Equilibrium in Repeated Games with Finite Automata," *Econometrica*, 56(6), 1259–1281.

Al-Najjar, N. I. (2009): "Decision Makers as Statisticians: Diversity, Ambiguity and Learning," *Econometrica*, 77(5), 1371–1401.

Al-Najjar, N. I., and M. M. Pai (2014): "Coarse decision making and overfitting," *J. Economic Theory*, 150, 467–486.

Arora, R., O. Dekel, and A. Tewari (2012): "Online Bandit Learning against an Adaptive Adversary: from Regret to Policy Regret," in *Proceedings of the 29th international coference on international conference on machine learning*, pp. 1747–1754.

Arora, R., M. Dinitz, T. Marinov, and M. Mohri (2018): "Policy Regret in Repeated Games," in *Proceedings of the 32nd international conference on neural information processing systems.*

Best, J., and D. Quigly (2020): "Persuasion for the Long Run," Discussion paper, Carnegie Mellon University and University of Oxford.

Blum, A., M. Hajiaghayi, K. Ligett, and A. Roth (2008): "Regret minimization and the price of total anarchy," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 373–382.

Braverman, M., J. Mao, J. Schneider, and M. Weinberg (2018): "Selling to a No-Regret Buyer," in *ACM Conf. on ACM Conference on Economics and Computation (ACM EC)*, pp. 523–538.

Brown, Z., and A. MacKay (2021): "Competition in Pricing Algorithms," .

Calvano, E., G. Calzolari, V. Denicolò, and S. Pastorello (2019): "Artificial Intelligence, Algorithmic Pricing and Collusion," .

Camara, M., J. Hartline, and A. Johnsen (2020): "Mechanisms for a No-Regret Agent: Beyond the Common Prior," *FOCS.*

Cherry, J., and Y. Salant (2019): "Statistical Inference in Games," Northwestern University.

Dembo, A., and O. Zeitouni (1998): *Large Deviations Techniques and Applications.* Springer-Verlag, New York, 2nd edn.

Deng, Y., J. Schneider, and B. Sivan (2019): "Strategizing against No-regret Learners," Discussion paper.

Dietterich, T. G. (2000): "Ensemble Methods in Machine Learning," in *Multiple Classifier Systems*, pp. 1–15, Berlin, Heidelberg. Springer Berlin Heidelberg.

Eliaz, K., and R. Spiegler (Forthcoming): "The Model Selection Curse," *American Economic Review: Insights.*

Foster, D., and R. Vohra (1997): "Calibrated Learning and Correlated Eequilibrium," *Games and Economic Behavior*, 21, 40–55.

Frenay, B., and M. Verleysen (2014): "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems,*, 25(5), 845–869.

Freund, Y. (2009): "A More Robust Boosting Algorithm," Discussion paper, University of California, San Diego.

Fudenberg, D., and D. Clark (2021): "Justified Communication Equilibrium," *American Economic Reivew*, Forthcoming.

Fudenberg, D., and K. He (2018): "Learning and Type Compatibility in Signaling Games," *Econometrica*, 86(4), 1215–1255.

Fudenberg, D., and D. M. Kreps (1995a): "Learning in Extensive Form Games I: Self-confirming Equilibria," *Journal of Economic Theory*, 8(1), 20–55.

——— (1995b): "Learning in Extensive Games, I: Self-Confirming and Nash Equilibrium," *Games and Economic Behavior*, 8, 20–55.

Fudenberg, D., and D. Levine (1998): *Learning in Games*. M.I.T. Press.

Fudenberg, D., and D. K. Levine (1993): "Steady State Learning and Nash Equilibrium," *Econometrica*, 61(3), 547–573.

——— (1995): "Consistency and Cautious Fictitious Play," *Journal of Economic Dynamics and Control*, 19, 1065–1089.

——— (2006): "Superstition and Rational Learning," *American Economic Reivew*, 96, 630–651.

Hart, S., and A. Mas-Colell (2000): "A Simple Adaptive Procedure Leading to Correlated Equilibrium," *Econometrica*, 68(5), 1127–1150.

Kamenica, E., and M. Gentzkow (2011): "Bayesian Persuasion," *American Economic Reivew*, 101(6), 2590–2615.

Liang, A. (2018): "Games of Incomplete Information Played by Statisticians," Discussion paper, University of Pennsylvania.

Long, P., and R. Servedio (2010): "Random Classification noise defeats all convex potential boosters," *Machine Learning*, 78(3), 287–304.

Maskin, E., and J. Tirole (1992): " The Principal-Agent Relationship with an Informed Principal, II: Common Values," *Econometrica*, 60(1), 1–42.

Meyn, S. P. (2007): *Control Techniques for Complex Networks*. Cambridge University Press.

Mukherjee, I., and R. E. Schapire (2013): "A Theory of Multiclass Boosting," *Journal of Machine Learning Research*, 14, 437–497.

Nekipelov, D., V. Syrgkanis, and E. Tardos (2015): "Econometrics for Learning Agents," in *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pp. 1–18.

Olea, J. L. M., P. Ortoleva, M. M. Pai, and A. Prat (2019): "Competing Models," Columbia University, Princeton University and Rice University.

Rambachan, A., J. Kleinberg, S. Mullainathan, and J. Ludwig (2020): "An Economic Approach to Regulating Algorithms," Discussion paper, Harvard Universitiy, Cornell University, and University of Chicago.

Rubinstein, A. (1986): "Finite Automata Play Repeated Prisoners Dilemma," *Journal of Economic Theory*, 39(1), 83–96.

——— (1993): "On Price Recognition and Computational Complexity in a Monopolistic Model," *Journal of Political Economy*, 101(3), 473–484.

Schapire, R. E., and Y. Freund (2012): *Boosting: Foundations and Algorithms*. MIT Press.

SHALEV-SHWARTZ, S., AND S. BEN-DAVID (2014): *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

SPENCE, A. M. (1973): “Job Market Signaling,” *Quarterly Journal of Economics*, 87(3), 355–374.

SPIEGLER, R. (2016): “ Bayesian Networks and Boundedly Rational Expectations *,” *The Quarterly Journal of Economics*, 131(3), 1243–1290.

ZHAO, C., S. KE, Z. WANG, AND S.-L. HSIEH (2020): “Behavioral Neural Networks,” Discussion paper.