

Learning in Online Principal-Agent Interactions: The Power of Menus

Minbiao Han¹, Michael Albert², Haifeng Xu¹

¹Department of Computer Science, The University of Chicago

²Darden Business School, University of Virginia

{minbiaohan, haifengxu}@uchicago.edu, albertm@darden.virginia.edu

Abstract

We study a ubiquitous learning challenge in online principal-agent problems during which the principal learns the agent’s private information from the agent’s revealed preferences in historical interactions. This paradigm includes important special cases such as pricing and contract design, which have been widely studied in recent literature. However, existing work considers the case where the principal can only choose a single strategy at every round to interact with the agent and then observe the agent’s revealed preference through their actions. In this paper, we extend this line of study to allow the principal to offer a *menu* of strategies to the agent and learn additionally from observing the agent’s selection from the menu. We provide a thorough investigation of several online principal-agent problem settings and characterize their sample complexities, accompanied by the corresponding algorithms we have developed. We instantiate this paradigm to several important design problems — including Stackelberg (security) games, contract design, and information design. Finally, we also explore the connection between our findings and existing results about online learning in Stackelberg games, and we offer a solution that can overcome a key hard instance of previous work.

1 Introduction

Asymmetric information is a key friction in many economic interactions. When setting a price for an item, the seller typically does not know the exact value the buyer has for the item. In hiring and compensation decisions, the employer does not know the work ethic of the particular employee. For a security professional, the specific value of different potential targets to an attacker may be unknown. Note that this information asymmetry is distinct from the standard strategic considerations in these interactions. For example, even if the security professional knows the value of all targets, an attacker may choose to attack a low-value target in order to avoid any potential defenses. However, if the security professional does not know the value of the targets to the attacker, it is much less likely that she will be able to implement a successful defense.

Learning from interactions where there is a single *principal* (referred to as she) who acts first and then an *agent*

(referred to as he) who responds, a class of problems denoted *principal-agent problems* (Myerson 1982; Gan et al. 2022), has generated significant interest in the computer science literature (e.g., Kleinberg and Leighton 2003; Letchford, Conitzer, and Munagala 2009; Haghtalab et al. 2022). However, existing work focuses on learning the agent’s type by using carefully chosen strategies from the possible set of principal strategies. In the pricing of a single item example, this would be the principal setting a price in every round of play to narrow down the agent’s value for the item. Specifically, the principal’s strategy space consists of offering a fixed price, p , and a probability of allocating an item, x , to the agent who has an unknown valuation v , which is his type. The agent can choose to purchase the item or not, providing information to the principal about the agent’s type. In this setting, the best strategy for the principal to learn the agent’s type exactly requires $O(\log(K))$ rounds, where K is the number of possible agent types.

Often, in practice, a principal does not choose a single strategy but instead offers a *menu* of strategies. The agent then chooses among the strategies, potentially revealing something about their type. Exploiting *revealed preferences* through offering a menu of strategies is common in pricing. For example, a retailer may bundle goods at different prices, or there may be different pricing based on the quantity purchased. If the principal can offer a menu of possible strategies, can she learn significantly more efficiently?

To illustrate the type of results we demonstrate in the revealed preferences setting with menus, consider the pricing problem again. In this setting, instead of posting a single price and allocation probability every round, the principal posts a menu of prices $\{x, p(x)\}_{x \in [0,1]}$. The agent then chooses a strategy, i.e. an x , from the menu, at which point the principal plays that strategy. In this case, the agent’s type can be learned in a single round. Specifically, if the payment function $p(x)$ is strictly convex, then $x^* = \operatorname{argmax}_x v \cdot x - p(x)$ is uniquely determined by v , and $v = \nabla p(x^*)$.

In this work, we examine, for general principal-agent problems, the question of the sample complexity of identifying the agent’s private type. We provide a condition, which for most common classes of principal-agent problems is *generic*, that allows the principal to identify the agent’s type in a *single round*. Additionally, for settings under which the aforementioned condition does not hold, we demonstrate

improvements using menus relative to state of the art single strategy results.

1.1 Related Work

Our setting is a generalized principal-agent problem (Myerson 1982; Gan et al. 2022). Online learning in principal-agent problems has been well studied in the context of pricing (Kleinberg and Leighton 2003; Amin, Rostamizadeh, and Syed 2013; Mohri and Munoz 2014; Dawkins, Han, and Xu 2021), Stackelberg games (Letchford, Conitzer, and Munagala 2009; Haghtalab et al. 2022), security games (Balcan et al. 2014; Peng et al. 2019), contract design problems (Ho, Slivkins, and Vaughan 2014; Zhu et al. 2022), and Bayesian persuasion (Castiglioni et al. 2020; Harris et al. 2023). There, the principal aims to learn the optimal strategy against an unknown agent through repeated interactions. The principal achieves this by posting a single principal strategy in each round and observing the corresponding response from the agent. Our results consider the setting where the principal can post (infinitely) many principal strategies. In Section 5.1, we provide direct comparisons to the learning environment of a single strategy, where we solve the previous hardness example by (Peng et al. 2019).

Another area of relevant literature is the research on learning economic parameters from revealed preferences (Beigman and Vohra 2006; Zadimoghaddam and Roth 2012; Balcan et al. 2014; Dawkins, Han, and Xu 2022). The main difference between this line of research and the previous research about online learning in principal-agent problems is that these works focus on learning the agent’s utility parameter, instead of the optimal principal strategy. Similar to us, they also consider learning the agent’s utility parameters. However, these previous works all focus on designing learning algorithms that post a single principal strategy at every round. In addition, we consider general principal-agent problems and instantiate the results to multiple specific problems, including the specific pricing problem considered in this literature.

More generally, our research aligns with the general field of learning from strategic data sources. This line of inquiry has been explored in various contexts and domains, addressing diverse objectives such as spam filtering (Brückner and Scheffer 2011), classification under incentive-compatibility constraints (Zhang and Conitzer 2021), and examining social implications (Akyol, Langbort, and Basar 2016; Milli et al. 2019). Finally, going beyond learning, our work subscribes to the line of information elicitation, which involves gathering relevant information from individuals to learn efficient algorithms. This literature is extensive and spans various disciplines, such as Bayesian persuasion (Dughmi and Xu 2017; Kamenica 2019), decision making (Savage 1971; Chen and Kash 2011), strategic data collection (Goel and Faltings 2020; Kong et al. 2020), and reverse Stackelberg games (Groot, De Schutter, and Hellendoorn 2012).

2 Preliminaries

We consider a generalized principal-agent problem (Myerson 1982; Gan et al. 2022). The one-round static version of

the problem can be characterized as a two-stage game with two players: the principal and the agent. The agent possesses a private type θ , which is randomly drawn from a finite set Θ with a prior probability distribution $\mu \in \Delta(\Theta)$. The principal is unaware of the specific realization of θ . Furthermore, the agent’s action is indicated by j , which is beyond the direct control of the principal. Let $[n] = \{1, 2, \dots, n\}$ denote the action space of the agent. The utility functions of the principal and the agent are denoted as $U(\cdot)$ and $V(\cdot)$, respectively, where $U/V : \mathcal{X} \times [n] \rightarrow \mathbb{R}$ and $\mathcal{X} \subseteq \mathbb{R}^m$ denotes the strategy space of the principal. Throughout this paper, we assume that the agent’s utility function $V(\cdot)$ is linear with respect to the principal strategy $\mathbf{x} \in \mathcal{X}$, that is,

$$V(\mathbf{x}, j) = \langle \mathbf{v}_j, \mathbf{x} \rangle + c_j, \quad (1)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, \mathbf{v}_j and c_j represent the agent’s utility parameters when he takes action j . We note that this assumption is widely adopted by the computer science literature studying principal-agent problems (Von Stengel and Zamir 2004; Dütting, Roughgarden, and Talgam-Cohen 2019; Gan et al. 2022). The utility information of distinct agent types is denoted by the superscript θ , that is, $V^\theta(\mathbf{x}, j) = \langle \mathbf{v}_j^\theta, \mathbf{x} \rangle + c_j^\theta$.

Throughout, vectors are denoted by bold lowercase letters (e.g., \mathbf{x}, \mathbf{v}_j). The i^{th} component of a vector is denoted in the subscript of a non-bold letter (e.g., $x_i, v_{j,i}$). In addition, we define a special relation between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$.

Definition 1. Given $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$, we denote $\mathbf{x} \parallel \mathbf{y}$ if there exists a scalar $\lambda \in \mathbb{R}$ such that $\mathbf{x} = \lambda \mathbf{y}$; otherwise, we denote $\mathbf{x} \not\parallel \mathbf{y}$.

Note that we always have $\mathbf{x} \parallel \mathbf{y}$ when $x, y \in \mathbb{R}$. Next, let us now consider some specific instantiations of the above generalized principal-agent problem framework.

Stackelberg Games. One widely adopted special case of generalized principal-agent problems is the celebrated model of Stackelberg games (Stackelberg 1934; Von Stengel and Zamir 2004; Gan et al. 2023). In such games, we commonly refer to the principal as the leader and the agent as the follower. The leader has m available actions and the follower has n available actions. The leader’s strategy can be represented by a randomized strategy $\mathbf{x} \in \Delta^m = \{\mathbf{x} : \sum_{i \in [m]} x_i = 1\}$, where x_i denotes the probability of playing each pure action i . The follower’s response is denoted by action $j \in [n]$. The leader/follower reward information is represented as matrix $L/F \in \mathbb{R}^{m \times n}$, where each element $L_{i,j}/F_{i,j}$ represents the leader/follower’s reward when the leader takes action i and the follower plays action j . As a result, we have $V^\theta(\mathbf{x}, j) = \sum_{i \in [m]} x_i F_{i,j}^\theta$, where F^θ represents the follower type θ ’s corresponding reward matrix.

Stackelberg Security Games. The Stackelberg Security Games (SSGs) have been studied for security resource allocation problems in many scenarios such as the Federal Air Marshal Service, the US Coast Guard, and the wildlife protection (Tambe 2011; An, Tambe, and Sinha 2017). The principal, known as the defender, has r resources to protect n targets from the attacker. Thus, the principal’s defense strategy can be represented by $\mathbf{x} \in \mathbb{R}^n$ where each x_t represents

the probability that target $t \in [n]$ is protected by the defender and $\sum_t x_t \leq r$. When a covered target $t \in [n]$ is attacked, the attacker gets penalty P_t^a and the defender gets reward R_t^d ; If an uncovered target is attacked, the attacker gets reward R_t^a and the defender gets penalty P_t^d . We assume the reward is preferred to the penalty, i.e., $R_t^d > P_t^d$ and $R_t^a > P_t^a$ for all t . The attacker's utility function is $V^\theta(\mathbf{x}, t) = x_t P_t^{a,\theta} + (1 - x_t) R_t^{a,\theta}$, where each type θ has a corresponding penalty/reward (i.e., $P_t^{a,\theta} / R_t^{a,\theta}$) for target t .

Contract Design Problems. Another instance of the principal-agent problem is the Nobel prize-winning research on contract theory (Hart and Holmström 1986). In a contract design problem, the agent has n actions that the principal cannot directly observe or control. Instead, the principal observes the m different possible outcomes and receives the reward $r_i \in \mathbb{R}$ associated with each outcome $i \in [m]$. We denote $\mathbf{r} \in \mathbb{R}^m$ as the reward vector for all outcomes. Each action of the agent $j \in [m]$ induces a certain distribution of outcomes $\mathbf{p}_j \in \Delta^m$ at cost c_j . The principal's strategy is to design a contract $\mathbf{x} \in \mathbb{R}^m$ that specifies the payment based on each possible outcome to incentivize the agent to take certain actions. As a result, we have $V^\theta(\mathbf{x}, j) = \sum_{i \in [m]} x_i p_{j,i}^\theta - c_j^\theta$, where each agent type θ has a corresponding cost c_j^θ for playing action j with a realization probability p_j^θ over the m outcomes.

Information Acquisition Games. In an information acquisition problem (Savage 1971; Li et al. 2022; Chen et al. 2023), we consider a stochastic environment with a principal and an agent. There exists a hidden state $w \in \Omega$ that influences the principal's utility but remains unknown to both the agent and the principal until the interaction ends. We denote $\Delta(\Omega)$ as a probabilistic belief over the hidden state. The principal initiates the process by offering a *scoring rule* $S : \Delta(\Omega) \times \Omega \rightarrow \mathbb{R}$ to the agent. The agent chooses an action $j \in [n]$ with a cost c_j and receives an observation $o \in O$ related to the hidden state w with probability $\Pr(o, w|j)$. They then provide a refined belief report $\sigma^o \in \Delta(\Omega)$ via the Bayesian rule (i.e., $\sigma^o(w) = \Pr(o, w|j) / \Pr(o|j)$), enabling the principal to make an informed decision. Finally, the hidden state w is revealed, and the principal determines the agent's payment based on the scoring rule. As a result, we have $V^\theta(\mathbf{x}, j) = \sum_{w,o} S(\sigma^o, w) \Pr^\theta(w, o|j) - c_j^\theta$, where $\sigma^o \in \Delta(\Omega)$, $w \in \Omega$, $\Pr^\theta(w, o|j)$ represents the agent θ 's probability of observing o if the hidden state is w and the action taken is j , and c_j^θ represents the corresponding cost.

3 When is a Single Round Sufficient?

In this section, we demonstrate the feasibility of learning the agent's private type within a single round, given the following assumption on the agent's finite type set Θ .

Assumption 3.1. *Given a finite-type set Θ and $\theta, \theta' \in \Theta$ such that $\theta \neq \theta'$, we assume that $\mathbf{v}_j^\theta \not\parallel \mathbf{v}_j^{\theta'}$ for every action $j \in [n]$, where $\mathbf{v}_j^\theta = \nabla V^\theta(\mathbf{x}, j)$.*

Note that this assumption is one of nondegeneracy. To be precise, each principal-agent problem instance is specified

by the principal and agent utility parameters. This can be thought of as a point in a high-dimensional space, one dimension for every parameter. As a result, we can think of every point in this high-dimensional space being a problem instance. The assumption holds with probability 1 since the game instances that violate the assumption in this space form a zero-measure set. Given this assumption, we are prepared to present the key findings of this section.

Theorem 1. *Suppose Θ satisfies assumption 3.1, we can construct a menu of principal strategies $\mathcal{M} \subseteq \mathcal{X}$ that learns the agent's private type $\theta \in \Theta$ in a single round.*

Proof. For every agent type θ , we can always separate the principal's strategy space \mathcal{X} into at most n sub-regions, where each sub-region consists of principal strategies that have the same agent best response (i.e., the agent response that maximizes his utility under the principal's strategy).

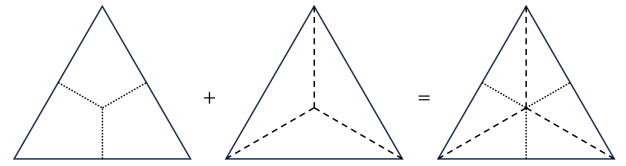


Figure 1: An example where $|\Theta| = 2$, $\mathcal{X} \subseteq \mathbb{R}^2$, and $n = 3$. For each specific agent type, the principal's strategy space can be divided into three sub-regions, where the agent's best response action remains the same at each sub-region. In this example, there exist six sub-regions where *all* agent types' best responses are fixed for principal strategies in each sub-region, though those fixed best responses could be different across different agent types.

Given a finite set Θ , we can pre-compute the separating hyper-planes from all types as in the example in Figure 1. There always exists a principal strategy $\hat{\mathbf{x}}$ that is an *interior* point such that all agent types' best responses are fixed around this principal strategy. Then the principal can post a menu of strategies centered at $\hat{\mathbf{x}}$ with a small radius ϵ (i.e., $\mathcal{M} = \{\mathbf{x} : |\mathbf{x} - \hat{\mathbf{x}}|^2 \leq \epsilon\} \subset \mathcal{X}$) such that all $\mathbf{x} \in \mathcal{M}$ have the same best response for every agent type. Such a menu \mathcal{M} always exists given a small enough ϵ and that $\hat{\mathbf{x}}$ is the *interior* point of some sub-region. Suppose the agent chooses $\mathbf{x}^* \in \mathcal{M}$ and responds with action j .

First, we can exclude those agent types whose best response with respect to $\hat{\mathbf{x}}$ is not j . As for the remaining possible types, we can simulate the agent's choice of $\mathbf{x}^* \in \mathcal{M}$ through the following optimization program.

$$\begin{aligned} \max_{\mathbf{x} \in \mathcal{M}} \quad & V^\theta(\mathbf{x}, j) \\ \text{s.t.} \quad & \sum_i (x_i - \hat{x}_i)^2 \leq \epsilon. \end{aligned} \quad (2)$$

Let us write the Lagrangian function of program (2) as $\mathcal{L}(\mathbf{x}, \lambda, \mu) = V^\theta(\mathbf{x}, j) + \lambda(\epsilon - \sum_i (x_i - \hat{x}_i)^2)$ and its corresponding Karush–Kuhn–Tucker (KKT) conditions:

$$\begin{aligned} v_{j,i}^\theta - 2\lambda x_i^* + 2\lambda \hat{x}_i &= 0, \quad \forall i \in [m] \\ \lambda(\epsilon - \sum_i (x_i^* - \hat{x}_i)^2) &= 0, \end{aligned}$$

$$\epsilon \geq \sum_i (x_i^* - \hat{x}_i)^2, \\ \mathbf{x}^*, \lambda \geq 0.$$

By solving the above KKT conditions, we have $x_i^*(\theta) = \frac{\sqrt{\epsilon} v_{j,i}^\theta}{\|v_j^\theta\|_2} + \hat{x}_i$, for all i . According to assumption 3.1, it follows that $v_j^\theta \not\parallel v_j^{\theta'}$, resulting in $\mathbf{x}^*(\theta) \neq \mathbf{x}^*(\theta')$ for all $\theta, \theta' \in \Theta$ and $\theta \neq \theta'$. As a result, we can learn the agent's type $\theta \in \Theta$ in a single round since each agent type θ has a unique choice of $\mathbf{x}^*(\theta)$ given our menu of principal strategies \mathcal{M} . \square

We remark that assumption 3.1 does not hold when $m = 1$, since we always have $v_j^\theta \parallel v_j^{\theta'}$ when $v_j^\theta, v_j^{\theta'} \in \mathbb{R}$. Thus, the menu of principal strategies \mathcal{M} we constructed for Theorem 1 does not hold for $m = 1$. Specifically, we have $x^*(\theta) = \sqrt{\epsilon} v_j^\theta / |v_j^\theta| + \hat{x} \in \{\hat{x} + \sqrt{\epsilon}, \hat{x} - \sqrt{\epsilon}\}$, which does not depend on θ , when $m = 1$. As a result, when $|\Theta| > 2$, there must exist two agent types that share the same preference for x^* when presented with the given menu of strategies \mathcal{M} , because $x^*(\theta)$ only has two possible choices. In Section 4, we will introduce effective algorithms that address this challenge and facilitate the learning of the agent's type specifically in scenarios where the principal action space is limited to one dimension. Next, we apply the generalized principal-agent framework to specific principal-agent problems.

3.1 Stackelberg Games

Our first instantiation is the Stackelberg game, where we show a corollary of Theorem 1. According to the definition of follower utility in a Stackelberg game, we have $v_{j,i}^\theta = \nabla V_i^\theta(\hat{\mathbf{x}}, j) = F_{i,j}^\theta$.

Corollary 1. *Suppose Θ satisfies assumption 3.1, we can construct a menu of leader strategies $\mathcal{M} \subseteq \mathcal{X}$ that learns the follower's private type $\theta \in \Theta$ in a single round.*

The proof is straightforward based on the proof of Theorem 1. However, a special aspect of Stackelberg games is that when $m = 2$, the leader's strategy space is $\mathcal{X} = \{\mathbf{x} : x_1 + x_2 = 1\} \subseteq \mathbb{R}$ and the follower's utility function is $v^\theta(\mathbf{x}, j) = (v_{j,1}^\theta - v_{j,2}^\theta)x_1 + v_{j,2}^\theta$. In other words, the actual dimension of the principal's strategy space is 1 when $m = 2$ in a Stackelberg game. In this case, Theorem 1 does not hold as we discussed before, and it is not possible to learn the follower's private type within a single round.

Proposition 1 (Impossibility of Single-round Learning). *There exists a finite set of follower types Θ such that it is impossible to learn the follower's type within a single round when the leader has 2 actions and $|\Theta| > 2$.*

Proof. We prove this by contradiction. Suppose the agent has K possible types (i.e., $\Theta = \{\theta_1, \dots, \theta_K\}$) and we can construct a menu of leader strategies \mathcal{M} such that there exists K unique $\{x^{\theta_k}\}_{\theta_k \in \Theta}$ where $x^{\theta_k} = \operatorname{argmax}_{\mathbf{x} \in \mathcal{M}} V^{\theta_k}(\mathbf{x})$. The agent's utility function $V^{\theta_k}(\mathbf{x}) = \max_j V^{\theta_k}(\mathbf{x}, j)$ is a piece-wise linear strictly convex function. By convexity, we have $\operatorname{argmax}_{\mathbf{x} \in \mathcal{M}} V^{\theta_k}(\mathbf{x}) \in \{\min(\mathcal{M}), \max(\mathcal{M})\}$, for all type θ_k , contradicting the fact there exists $k > 2$ unique choices of all follower types. \square

3.2 Contract Design Problems

We also consider the special contract design problem where $v_{j,i}^\theta = \nabla V_i^\theta(\mathbf{x}, j) = p_{j,i}^\theta$, which represents the probability to achieve outcome $i \in [m]$ when the agent takes action $j \in [n]$. Next, we apply Theorem 1 and show a corollary result for the contract design problem.

Corollary 2. *Suppose Θ satisfies assumption 3.1, we can construct a menu of principal contracts $\mathcal{M} \subseteq \mathcal{X}$ that learns the agent's private type $\theta \in \Theta$ in a single round.*

3.3 Information Acquisition Games

Another special case of the generalized principal-agent problem is the information acquisition problem. From the definition of agent utility function, we have $v_{j,i}^\theta = \nabla V_i^\theta(\mathbf{x}, j) = \Pr^\theta(w, o|j)$ where i denotes a pair of variables $(w, o), w \in \Omega, o \in O$. Thus, v_j^θ in the information acquisition game represents a vector of length $|\Omega| \times |O|$. We can also apply Theorem 1 for the following corollary.

Corollary 3. *Suppose Θ satisfies assumption 3.1, we can construct a menu of principal scoring rules $\mathcal{M} \subseteq \mathcal{X}$ that learns the agent's private type $\theta \in \Theta$ in a single round.*

3.4 Stackelberg Security Games

Finally, we consider the specific Stackelberg security game. Recall that the attacker's utility to attack target j only depends on the defender's coverage probability of protecting target j , that is,

$$V^\theta(\mathbf{x}, j) = P_j^{a,\theta} x_j + (1 - x_j) R_j^{a,\theta}. \quad (3)$$

As a result, the actual dimension of the defender's strategy space in $V^\theta(\mathbf{x}, j)$ is 1 (i.e., x_j the probability of defending target j), and the result from Theorem 1 cannot be applied. Intuitively, when presented with a menu of defender strategies, all agent types would favor the one with the lowest protection probability for target j , given that their best response with respect to all strategies from this menu is fixed to attack this particular target j . The underlying reason is $v_{j,i}^\theta = \nabla V^\theta(\mathbf{x}, j) = P_j^{a,\theta} - R_j^{a,\theta} \in \mathbb{R}$. As a result, assumption 3.1 cannot hold for more than two attacker types in a security game, and Theorem 1 is also not applicable.

4 When is a Small Number of Rounds Sufficient?

As discussed, our single-round results do not hold when the effective dimensionality of the principal's strategy space is 1 (see Proposition 1). In the following section, we introduce algorithms designed to learn the private agent type efficiently with a small number of rounds in this particular scenario, i.e., $\mathcal{X} \subseteq \mathbb{R}$. For the sake of simplicity, we can assume that $\mathcal{X} = [0, 1]$, as the principal's strategy can always be rescaled within this range without loss of generality.

First, we present an algorithm for the principal to learn the agent's type within $\log |\Theta|$ rounds under the following assumption on the finite type set Θ .

Assumption 4.1. Given a finite type set Θ , we assume for every type $\theta \in \Theta$, there does not exist $j \in [n]$ such that

$$V^\theta(x, j) \geq V^\theta(x, j'), \quad \forall x, j' \neq j \quad (4)$$

Assumption 4.1 is a statement that no agent type has a dominant action. This ensures there is sufficient variation in the agent’s behavior for the principal to learn the agent’s type.

Theorem 2. When $m = 1$ and Θ satisfies assumption 4.1, there exists an algorithm to learn the agent’s type in $\log |\Theta|$ rounds where the principal posts a menu of principal strategies per round.

Proof. We propose the following algorithm 1 to learn the agent’s true type $\theta \in \Theta$ in $\log |\Theta|$ rounds.

Algorithm 1: LEARNING-VIA-MENU

Input: Agent’s type set: $\Theta = \{\theta_1, \dots, \theta_K\}$

Output: Agent’s true type θ

- 1: Let $V^{\theta_k}(x) = \max_j V^{\theta_k}(x, j), \forall \theta_k \in \Theta$.
 - 2: Compute $x^{\theta_k} = \operatorname{argmin}_x V^{\theta_k}(x), \forall \theta_k \in \Theta$, let $\mathbf{x} = \operatorname{sorted}([x^{\theta_1}, \dots, x^{\theta_K}])$
 - 3: **while** $\operatorname{len}(\mathbf{x}) > 1$: **do**
 - 4: $n = \operatorname{len}(\mathbf{x}); \operatorname{mid} = \lfloor \frac{n}{2} \rfloor; x_1, x_2 = \mathbf{x}[\operatorname{mid}], \mathbf{x}[\operatorname{mid}+1]$
 - 5: Provide a menu $\mathcal{M} = (x_1, x_2)$ of two principal strategies to the agent
 - 6: **if** the agent chooses x_1 **then**
 - 7: $\mathbf{x} = \mathbf{x}[\operatorname{mid} + 1 :]$ (deleting all the elements $x[0], \dots, x[\operatorname{mid}]$ from \mathbf{x})
 - 8: **else**
 - 9: $\mathbf{x} = \mathbf{x}[: \operatorname{mid}]$ (deleting all the elements $x[\operatorname{mid} + 1], \dots, x[n - 1]$ from \mathbf{x})
 - 10: **end if**
 - 11: **end while**
 - 12: **return** the only x^{θ_k} in \mathbf{x} , where θ_k is the agent’s type
-

The correctness of the proof of this algorithm relies on a key insight that $V^{\theta_k}(x)$ is a convex function since it is the maximum over a set of linear functions. Thus, given two principal strategies x_1, x_2 where $x_1 < x_2$, if the agent θ prefers x_1 , then the agent’s $x^\theta = \operatorname{argmin}_x V^\theta(x)$ must be on the right of x_1 by the convex property of agent’s utility function. A similar argument holds when the agents prefer x_2 . As a result, in each round we can eliminate half of the types from the agent’s type set, leading to a $\log |\Theta|$ sample complexity.

However, there is still one more case we need to consider. When two agent types $\theta_k, \theta_{k'}$ have the same $x^{\theta_k} = x^{\theta_{k'}}$ in Line 2, they will always choose the same strategy for any menu of two strategies from Line 4. In this case, the principal can distinguish these two agent types by observing the agent’s corresponding response to the chosen principal’s strategy. By assumption 4.1 there does not exist a dominant action for any agent type. Therefore, every agent type has more than one best response at x^{θ_k} . We follow the convention that when the agent is indifferent between multiple responses, we can specify the tie-breaking rule (Letchford, Conitzer, and Munagala 2009; Peng et al. 2019), ensuring that different types respond with different actions. Thus,

the principal can still distinguish these different agent types according to different responses specified by different tie-breaking rules. \square

What is more, we also consider the scenario where the principal cannot offer a menu of principal strategies, but only a *single* principal strategy. This setting has been extensively studied in the literature where $m > 1$ (Letchford, Conitzer, and Munagala 2009; Blum, Haghtalab, and Procaccia 2014; Peng et al. 2019). Next, we show an efficient algorithm when $m = 1$, which can learn the agent’s private type in $\log |\Theta|$ rounds as our previous results. Though our next algorithm achieves the same $\log |\Theta|$ complexity as our previous result in Theorem 2, it requires the following additional assumption on the finite agent type set Θ since the principal now has less flexibility and can only post a *single* strategy per round.

Assumption 4.2. Given a finite type set Θ , we assume there does not exist $\theta, \theta' \in \Theta, x \in \mathcal{X}$ and $j, j', j'' \in [n]$ such that

$$V^\theta(x, j) = V^\theta(x, j') \text{ and } V^{\theta'}(x, j) = V^{\theta'}(x, j'') \quad (5)$$

where $\theta \neq \theta', j \neq j',$ and $j \neq j''$.

At a high level, assumption 4.2 requires that there do not exist two agent types who not only have the same breakpoint in their piece-wise linear convex utility functions but also change from the same best response to different best responses at this breakpoint. In Figure 2, we provide an illustration of the utility functions associated with two agent types. It is important to observe that instances failing to satisfy assumption 4.2 require not only the breakpoints of two piece-wise linear convex functions to coincide, which forms a zero measure set in the entire strategy space, but also necessitate the transition from the same response to different responses for these piecewise linear functions (e.g., $j_1^\theta = j_1^{\theta'}$ in Figure 2). Hence, we emphasize that assumption 4.2 is a nondegeneracy assumption since any randomly generated instance will satisfy this assumption with probability 1.

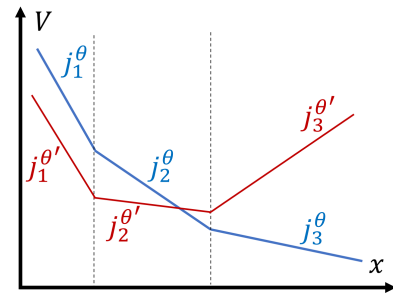


Figure 2: An example of agents’ utility functions V^θ and $V^{\theta'}$ when principal strategy $x \in \mathbb{R}$.

Theorem 3. When $m = 1$ and Θ satisfies assumptions 4.1 & 4.2, there exists an algorithm to learn the agent’s type in $\log |\Theta|$ rounds where the principal only posts a single strategy per round.

Proof. We propose the following algorithm 2 to learn the agent’s true type in $\log |\Theta|$ rounds.

Algorithm 2: LEARNING-VIA-SINGLE-STRATEGY

Input: Agent's type set: $\Theta = \{\theta_1, \dots, \theta_K\}$
Output: Agent's true type θ

- 1: Let $V^{\theta_k}(x) = \max_j V^{\theta_k}(x, j), \forall \theta_k \in \Theta$.
- 2: **while** $|\Theta| > 1$: **do**
- 3: For all $j \in [n]$, compute $n_{x=0}^j$ as the number of agent types who best respond to $x = 0$ with j .
- 4: Denote $n_{x=0}^{j^*} = \max_j n_{x=0}^j$.
- 5: **if** $n_{x=0}^{j^*} > \lfloor \frac{|\Theta|}{2} \rfloor$ **then**
- 6: Compute $x^* \in (0, 1]$ such that $n_{x=x^*}^{j^*} = \lfloor \frac{|\Theta|}{2} \rfloor$
- 7: Principal plays strategy x^* :
- 8: **if** Agent responds with j^* **then**
- 9: Remove agent types whose best response to x^* is not j^* from Θ
- 10: **else**
- 11: Remove agent types whose best response to x^* is j^* from Θ
- 12: **end if**
- 13: **else**
- 14: Principal plays strategy $x^* = 0$:
- 15: **if** Agent responds with j^* **then**
- 16: Remove agent types whose best response to x^* is not j^* from Θ
- 17: **else**
- 18: Agent responds with j'
- 19: Remove agent types whose best response to x^* is not j' from Θ
- 20: **end if**
- 21: **end if**
- 22: **end while**
- 23: **return** Θ

The proof of this algorithm relies on a key insight that when $n_{x=0}^{j^*} > \frac{|\Theta|}{2}$, we can always find a principal strategy x^* such that the number of agent types responding to x^* with j^* is $\lfloor \frac{|\Theta|}{2} \rfloor$ (i.e. Line 6). The reason is as follows. Given any agent action $j \in [n]$, we denote $n_{x=0}^j$ and $n_{x=1}^j$ as the number of agent types who respond to $x = 0$ and $x = 1$ (i.e., the left-most and right-most principal strategy in principal's strategy space) with action j .

According to assumption 4.1, there does not exist a dominated action for every agent type θ . Thus, we must have $n_{x=0}^j + n_{x=1}^j \leq |\Theta|$. In addition, by assumption 4.2, there do not exist two agent types that change from the same agent response to different responses at the same principal strategy. As a result, we must have $n_{x=0}^j$ always changes smoothly (i.e., changes at most by 1) to $n_{x=1}^j$. Therefore, if $n_{x=0}^{j^*} > \lfloor \frac{|\Theta|}{2} \rfloor$ in Algorithm 2, then we must have $n_{x=1}^{j^*} < \lfloor \frac{|\Theta|}{2} \rfloor$ and there must exist $x^* \in (0, 1)$ such that the number of agent types responding to x^* with j^* is exactly $\lfloor \frac{|\Theta|}{2} \rfloor$.

On the other hand (i.e., Line 13), if $n_{x=0}^{j^*} \leq \lfloor \frac{|\Theta|}{2} \rfloor$, then we know there already exists such an $x^* = 0$. This is due to the fact that for every possible agent response action $j \in [n]$, the number of agent types that responds to $x^* = 0$ with this

action is less than or equal to $\lfloor \frac{|\Theta|}{2} \rfloor$. Thus, we can remove at least $\lfloor \frac{|\Theta|}{2} \rfloor$ possible types no matter which response is observed when playing principal strategy $x^* = 0$.

As a result, each round in the **while** loop (line 2) is guaranteed to remove at least $\lfloor \frac{|\Theta|}{2} \rfloor$ possible agent types from the Θ set, leading to a $\log |\Theta|$ rounds learning algorithm. \square

Finally, as we discussed in Section 3.4, the actual dimension of the defender's strategy space is 1 in Stackelberg security games. Similarly, we can apply Algorithm 2 to learn the attacker's private type in $\log |\Theta|$ rounds.

Corollary 4. *Suppose Θ satisfies assumptions 4.1 & 4.2, there exists an algorithm to learn the attacker's private type in $\log |\Theta|$ rounds.*

The proof is straightforward based on Theorem 3 and Algorithm 2. The role of the defender and attacker corresponds to the principal and agent in the Stackelberg game. Similarly, Algorithm 2 can be applied to security games.

5 When are Polynomially Many Rounds Sufficient?

In this section, we consider the more general case where the principal does not have prior information (i.e., Θ) about the agent's private type, and this private type can be drawn from an *infinite* type space (see Letchford, Conitzer, and Munagala (2009); Blum, Haghtalab, and Procaccia (2014); Peng et al. (2019) for previous work without menus). Following a similar argument to Theorem 1, we can construct a menu of strategies centered at \hat{x} with a small radius ϵ (i.e., $\mathcal{M} = \{x : |x - \hat{x}|^2 \leq \epsilon\}$). Agent θ 's best response to all principal strategies within this menu is $j \in [n]$ and we denote the agent's optimal choice as $x^*(\theta)$. From the analysis of Theorem 1, we have

$$x_i^*(\theta) = \sqrt{\epsilon} v_{j,i}^\theta / \|v_j^\theta\|_2 + \hat{x}_i, \quad \forall i \in [m]. \quad (6)$$

As a result, from equation (6) we have

$$v_{j,i}^\theta / \|v_j^\theta\|_2 = (x_i^*(\theta) - \hat{x}_i) / \sqrt{\epsilon}, \quad \forall i \in [m]. \quad (7)$$

From the above m equations (7), we can compute v_j^θ up to a factor of $\lambda_j \in \mathbb{R}$, i.e., $v_j^\theta = \lambda_j \tilde{v}_j^\theta$ where \tilde{v}_j^θ is a *unit vector* that satisfies (7). As a result, learning the private agent utility parameters $v_j^\theta \in \mathbb{R}^m$ is reduced to learning $\lambda_j \in \mathbb{R}$.

Before proceeding to our main result, we assume that there is an oracle that provides the principal with a set of strategies that induce the agent to best respond with all possible actions for some strategy in the set.

Assumption 5.1 (Agent Action-informed Oracle). *There exists an oracle that provides n principal strategies $\{\hat{x}_j\}_{j \in [n]}$ such that under principal strategy \hat{x}_j , the agent's best response is action j .*

This assumption has been adopted by previous work (Chen et al. 2023; Zhao et al. 2023). It is considerably less stringent compared to existing online learning models in strategic environments, which assumes that the principal can anticipate the agent's best response or possesses knowledge of certain parameters of the agent's utility function (Chajewska, Koller, and Ormoneit 2001; Cesa-Bianchi and Lugosi

2006; Shalev-Shwartz et al. 2012). In addition, this may be a reasonable assumption in practice for certain classes of principal-agent problems. For example, in the contract design problem, each outcome may be achieved primarily by a specific action (e.g., a high-quality outcome is most likely achieved by an action with high effort). Therefore, we can induce one specific action by rewarding its corresponding outcome while setting the reward for other outcomes as 0. With assumption 5.1, we are now ready to present the main result of this section.

Theorem 4. *Under assumption 5.1, the agent’s private type can be learned with $O(n^2L)$ rounds, where L is the representation precision.*

Proof Sketch. The high-level proof idea is as follows. Given any menu $\mathcal{M}_j = \{\mathbf{x} : |\mathbf{x} - \hat{\mathbf{x}}_j|^2 \leq \epsilon\}$ for all $j \in [n]$, we can learn the agent’s utility parameters \mathbf{v}_j^θ up to a factor of λ_j , that is, $\mathbf{v}_j^\theta = \lambda_j \tilde{\mathbf{v}}_j^\theta$ where $\tilde{\mathbf{v}}_j^\theta$ is a unit vector that can be computed by (7). Therefore, learning the agent’s utility function (i.e., \mathbf{v}_j^θ, c_j for all $j \in [n]$) is reduced to learning λ_j, c_j for all $j \in [n]$. Furthermore, for any principal strategy $\hat{\mathbf{x}}_{j,j'}$ that’s on the separating hyperplane between the best response sub-regions of two actions $j, j' \in [n]$ (see our Figure 1 for an example of dividing the principal’s strategy space into multiple best response sub-regions), we have the agent’s utility of responding with j equals to the utility of responding with j' under the principal strategy $\hat{\mathbf{x}}_{j,j'}$, that is,

$$\lambda_j \langle \tilde{\mathbf{v}}_j, \hat{\mathbf{x}}_{j,j'} \rangle + c_j = \lambda_{j'} \langle \tilde{\mathbf{v}}_{j'}, \hat{\mathbf{x}}_{j,j'} \rangle + c_{j'}. \quad (8)$$

where $\lambda_j, \lambda_{j'}, c_j$, and $c_{j'}$ are unknown variables. Thus, sampling 4 principal strategies on this hyperplane is enough to learn these four variables. We continue this process for all possible hyperplanes, and there are at most n^2 in total. Detailed proof can be seen in the appendix. \square

It is still an open question whether it is sufficient to learn with fewer than $O(n^2L)$ rounds. Using menus, our sample complexity of $O(n^2L)$ is an improvement over the current state of the art, $O(n^3L)$ (Peng et al. 2019), for settings where the principal can only offer a single strategy, an improvement of $O(n)$.

5.1 Online Learning in Stackelberg Games

In this subsection, we demonstrate that the ability to offer a menu can dramatically improve the sample complexity of a well-known hardness example from Peng et al. (2019) for Stackelberg games. To begin, we provide a summary of current results on Stackelberg games as in Table 1.

Number of rounds	$m = 2$	$m > 2$
Menu	$\log \Theta $ Theorem 2	$O(1)$ Corollary 1
Single strategy	$\log \Theta $ Theorem 3	$ \Theta $ (Peng et al. 2019)

Table 1: Summary of the results discussed regarding the required number of interaction rounds to learn the private follower’s type from a finite type set Θ in Stackelberg games.

Example 1 (Lemma 8, (Peng et al. 2019)). *Consider the Stackelberg game, where the leader has m actions and the follower has $n = m + 2$ actions. The utility parameters of every follower type θ is*

$$F_{\cdot,j}^\theta = \left(\underbrace{-\frac{2}{m-2}, \dots, -\frac{2}{m-2}}_{i-1}, 1, \underbrace{-\frac{2}{m-2}, \dots, -\frac{2}{m-2}}_{m-i} \right), \forall j \in [m];$$

$$F_{\cdot,m+1}^\theta = \left(\underbrace{-\frac{1}{N^3}, -\frac{1}{N^3}, \dots, -\frac{1}{N^3}}_m \right);$$

$$F_{\cdot,a^*}^\theta = \frac{1}{N^2} \mathbf{v}^\theta.$$

where $\theta \subseteq [m]$ with $|\theta| = m/2$ or 0, that is, θ is either an empty set or a set with $m/2$ elements that are drawn from set $[m] = \{1, \dots, m\}$. $\mathbf{v}^\theta \in \{1, -N\}^m$ is a vector of length m whose i -th element is 1 if $i \in \theta$ and $-N$ otherwise. There are $\binom{m}{m/2} + 1$ follower types in total.

In the case of this Example 1 from Peng et al. (2019), they show that a minimum of $2^{\Omega(m)}$ samples is necessary for learning the follower’s private type from $\Theta = \emptyset \cup \{\theta : |\theta| = m/2 \text{ and } \theta \subset [m]\}$. We demonstrate the follower’s type can be learned within a *single* round with menus.

Proposition 2. *Within a single round, a menu of leader strategies is available to learn the follower’s private type in Example 1.*

In our proof, we construct a menu of leader strategies where every different follower type prefers different leader strategies. We refer the reader to the detailed construction of this menu in the appendix.

6 Conclusion

Motivated by the recent research on the power of menus in contract design (Gan et al. 2022; Castiglioni, Marchesi, and Gatti 2022; Guruganesh et al. 2023), this paper put forwards a variant of the very basic online learning model in principal-agent problems by augmenting the principal policy space with posting a menu of principal strategies instead of a single principal strategy at every round of online interaction with the agent. We provide an understanding of the learning complexities for a variety of problem settings, offering dedicated learning algorithms. Furthermore, we apply our general principal-agent framework to various concrete game instances, including widely studied cases like the contract design problem and the Stackelberg game, establishing a connection between our results and earlier online learning results in the literature.

There are many open questions around the power of menus in online principal-agent problems. For example, while our focus is to learn the agent’s private type in this paper, it would be interesting to explore the problem of optimizing the principal utility throughout the online principal-agent interactions using menus. What is more, our model assumes no constraints on the size of principal menus. A natural question would be characterizing the sample complexity if the principal can only post a finite menu of strategies whose size is upper bound by a constant.

Acknowledgments

This work is supported by NSF award CCF-2303372, Army Research Office Award W911NF-23-1-0030 and Office of Naval Research Award N00014-23-1-2802.

References

- Akyol, E.; Langbort, C.; and Basar, T. 2016. Price of Transparency in Strategic Machine Learning. *arXiv*, arXiv-1610.
- Amin, K.; Rostamizadeh, A.; and Syed, U. 2013. Learning prices for repeated auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, 1169–1177.
- An, B.; Tambe, M.; and Sinha, A. 2017. Stackelberg security games (ssg) basics and application overview. *Improving Homeland Security Decisions*, 485.
- Balcan, M.-F.; Daniely, A.; Mehta, R.; Urner, R.; and Vazirani, V. V. 2014. Learning economic parameters from revealed preferences. In *International Conference on Web and Internet Economics*, 338–353. Springer.
- Beigman, E.; and Vohra, R. 2006. Learning from revealed preference. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, 36–42.
- Blum, A.; Haghtalab, N.; and Procaccia, A. D. 2014. Learning optimal commitment to overcome insecurity. In *Advances in Neural Information Processing Systems*, 1826–1834.
- Brückner, M.; and Scheffer, T. 2011. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 547–555.
- Castiglioni, M.; Celli, A.; Marchesi, A.; and Gatti, N. 2020. Online bayesian persuasion. *Advances in Neural Information Processing Systems*, 33: 16188–16198.
- Castiglioni, M.; Marchesi, A.; and Gatti, N. 2022. Designing Menus of Contracts Efficiently: The Power of Randomization. *arXiv preprint arXiv:2202.10966*.
- Cesa-Bianchi, N.; and Lugosi, G. 2006. *Prediction, learning, and games*. Cambridge university press.
- Chajewska, U.; Koller, D.; and Ormoneit, D. 2001. Learning an agent’s utility function by observing behavior. In *ICML*, 35–42.
- Chen, S.; Wu, J.; Wu, Y.; and Yang, Z. 2023. Learning to Incentivize Information Acquisition: Proper Scoring Rules Meet Principal-Agent Model. In *Proceedings of the 40th International Conference on Machine Learning*.
- Chen, Y.; and Kash, I. 2011. Information elicitation for decision making.
- Dawkins, Q.; Han, M.; and Xu, H. 2021. The Limits of Optimal Pricing in the Dark. *Advances in Neural Information Processing Systems*, 34: 26649–26660.
- Dawkins, Q.; Han, M.; and Xu, H. 2022. First-Order Convex Fitting and Its Application to Economics and Optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6480–6487.
- Dughmi, S.; and Xu, H. 2017. Algorithmic persuasion with no externalities. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, 351–368.
- Dütting, P.; Roughgarden, T.; and Talgam-Cohen, I. 2019. Simple versus optimal contracts. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, 369–387.
- Gan, J.; Han, M.; Wu, J.; and Xu, H. 2022. Optimal Coordination in Generalized Principal-Agent Problems: A Revisit and Extensions. *arXiv preprint arXiv:2209.01146*.
- Gan, J.; Han, M.; Wu, J.; and Xu, H. 2023. Robust Stackelberg Equilibria. In *Proceedings of the 24th ACM Conference on Economics and Computation*, EC ’23, 735. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701047.
- Goel, N.; and Faltings, B. 2020. Personalized peer truth serum for eliciting multi-attribute personal data. In *Uncertainty in Artificial Intelligence*, 18–27. PMLR.
- Groot, N.; De Schutter, B.; and Hellendoorn, H. 2012. Reverse stackelberg games, part i: Basic framework. In *2012 IEEE International Conference on Control Applications*, 421–426. IEEE.
- Guruganesh, G.; Schneider, J.; Wang, J.; and Zhao, J. 2023. The Power of Menus in Contract Design. *arXiv preprint arXiv:2306.12667*.
- Haghtalab, N.; Lykouris, T.; Nietert, S.; and Wei, A. 2022. Learning in Stackelberg Games with Non-myopic Agents. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, 917–918.
- Harris, K.; Immorlica, N.; Lucier, B.; and Slivkins, A. 2023. Algorithmic Persuasion Through Simulation: Information Design in the Age of Generative AI. *arXiv preprint arXiv:2311.18138*.
- Hart, O. D.; and Holmström, B. 1986. The theory of contracts.
- Ho, C.-J.; Slivkins, A.; and Vaughan, J. W. 2014. Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. In *Proceedings of the fifteenth ACM conference on Economics and computation*, 359–376.
- Kamenica, E. 2019. Bayesian persuasion and information design. *Annual Review of Economics*, 11: 249–272.
- Kleinberg, R.; and Leighton, T. 2003. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, 594–605. IEEE.
- Kong, Y.; Schoenebeck, G.; Tao, B.; and Yu, F.-Y. 2020. Information elicitation mechanisms for statistical estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 2095–2102.
- Letchford, J.; Conitzer, V.; and Munagala, K. 2009. Learning and approximating the optimal strategy to commit to. In *International Symposium on Algorithmic Game Theory*, 250–262. Springer.
- Li, Y.; Hartline, J. D.; Shan, L.; and Wu, Y. 2022. Optimization of Scoring Rules. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC ’22, 988–989.

New York, NY, USA: Association for Computing Machinery. ISBN 9781450391504.

Milli, S.; Miller, J.; Dragan, A. D.; and Hardt, M. 2019. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 230–239.

Mohri, M.; and Munoz, A. 2014. Optimal regret minimization in posted-price auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, 1871–1879.

Myerson, R. B. 1982. Optimal coordination mechanisms in generalized principal–agent problems. *Journal of mathematical economics*, 10(1): 67–81.

Peng, B.; Shen, W.; Tang, P.; and Zuo, S. 2019. Learning optimal strategies to commit to. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2149–2156.

Savage, L. J. 1971. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336): 783–801.

Shalev-Shwartz, S.; et al. 2012. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2): 107–194.

Stackelberg, H. v. 1934. Marktform und gleichgewicht.

Tambe, M. 2011. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge university press.

Von Stengel, B.; and Zamir, S. 2004. Leadership with commitment to mixed strategies. Technical report, Citeseer.

Zadimoghaddam, M.; and Roth, A. 2012. Efficiently learning from revealed preference. In *International Workshop on Internet and Network Economics*, 114–127. Springer.

Zhang, H.; and Conitzer, V. 2021. Incentive-aware PAC learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 5797–5804.

Zhao, G.; Zhu, B.; Jiao, J.; and Jordan, M. 2023. Online Learning in Stackelberg Games with an Omniscient Follower. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 42304–42316. PMLR.

Zhu, B.; Bates, S.; Yang, Z.; Wang, Y.; Jiao, J.; and Jordan, M. I. 2022. The Sample Complexity of Online Contract Design. *arXiv preprint arXiv:2211.05732*.