# Gaming Helps!
# Learning from Strategic Interactions in Natural Dynamics

**Yahav Bechavod**
Hebrew University

**Katrina Ligett**
Hebrew University

**Zhiwei Steven Wu**
Carnegie Mellon University

**Juba Ziani**
University of Pennsylvania

## Abstract

We consider an online regression setting in which individuals adapt to the regression model: arriving individuals are aware of the current model, and invest strategically in modifying their own features so as to improve the predicted score that the current model assigns to them. Such feature manipulation has been observed in various scenarios—from credit assessment to school admissions—posing a challenge for the learner. Surprisingly, we find that such strategic manipulations may in fact help the learner recover the meaningful variables—that is, the features that, when changed, affect the true label (as opposed to non-meaningful features that have no effect). We show that even simple behavior on the learner's part allows her to simultaneously i) accurately recover the meaningful features, and ii) incentivize agents to invest in these meaningful features, providing incentives for improvement.

## 1 Introduction

As algorithmic decision-making takes a more and more important role in myriad application domains, incentives emerge to change the inputs presented to these algorithms. Recently, a collection of very interesting papers has explored various models of strategic behavior on the part of the classified individuals in learning settings, and ways to mitigate the harms to accuracy that can arise from falsified features [6, 1, 11, 8]. Additionally, some recent work has focused on the design of learning algorithms that incentivize the classified individuals to make "good" investments in true changes to their variables [15].

The present paper takes a different tack, and explores another potential effect of strategic investment in true changes to variables, in an online learning setting: we claim that interaction between the online learning and the strategic individuals may actually aid the learning algorithm in identifying *meaningful* variables. By meaningful, we mean, informally, and within the context of this paper, variables for which changing their true value affects the true label and thus, may lead agents to improve. In contrast, *non-meaningful* variables do not affect the true label; such features are susceptible to gaming, as they can potentially be used to obtain better outcomes with respect to the posted model without actually improving true labels.

The idea is quite simple. First, if a learning algorithm's hypothesis at a particular round depends heavily on a certain variable, this incentivizes the arriving individual to invest in improving that variable. If that variable were meaningful (that is, it has an effect on the true label), then the learner would observe an improved true label, increasing the observed correlation between the variable and the label. However, if that variable were non-meaningful, the changes would not have an effect on the true label, reducing the observed correlation between the variable and the label. Second, if a learning algorithm improves its hypotheses over time, this changing sequence of incentives should encourage investment in a variety of promising variables, exposing those that are meaningful. This process should naturally induce the learner to shift its dependence towards meaningful variables, thereby incentivizing individuals to invest in improving as opposed to gaming, resulting in an overall higher-quality

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. 

population.

The goal of this paper is to highlight this potential beneficial effect of the interaction between online learning and strategic modification. To do so, we choose to focus our study on a simple linear regression setting. In our model, there is a true underlying latent regression parameter vector $\beta^*$, and there is an underlying distribution over unmodified feature vectors. On every round $t$, the learner must announce a regression vector $\hat{\beta}_t$.[1] An individual then appears, with an unmodified feature vector $x_t$ chosen i.i.d. from the distribution. Before presenting himself to the learner, the individual observes $\hat{\beta}_t$ and has the opportunity to invest in changing his true features to some $\bar{x}_t$; we focus on a simple model wherein the individual's investment results in a targeted change to a single variable. The individual then receives utility $\langle \hat{\beta}^t, \bar{x}_t \rangle$, and the learner gets feedback $\bar{y}_t = \beta^{*\intercal} \bar{x}_t + \varepsilon_t$, where $\varepsilon_t$ is some noise.

Within this simple model, we consider simple behaviors for both the learner and the individuals: At each time $t$, the individual modifies his features so as to maximize his utility given the posted $\hat{\beta}_t$; periodically, the learner updates $\hat{\beta}_t$ with her best estimate of $\beta^*$ given the (modified) features and labels she has observed, via least-square regression. Our main result is that under this simple behavior, the learner recovers $\beta^*$ accurately, after observing sufficiently many individuals. Our result is divided in two parts: first, we show that least-square regression accurately recovers $\beta^*$ with respect to features that many individuals have invested in. Second, we show that these dynamics incentivize investments in every feature, leading to accurate recovery of $\beta^*$ in its entirety, under an assumption on how the learner breaks ties between multiple least-square solutions. Our accuracy guarantees for a feature improve with the number of times that feature is invested in.

It is important to emphasize that we focus on a setting in which individuals' modifications (which we refer to interchangeably as "manipulations") of their variables can be true investments (e.g., studying to achieve better mastery of material before an exam—the exam score is the variable and the mastery level is the label) rather than deceitful manipulations (e.g., cheating on the exam to achieve a higher score without improving mastery). Deceitful manipulations would not help to expose meaningful variables, because such changes would never affect the true label (subject mastery), regardless of whether the manipulations were in meaningful or non-meaningful variables.

[1]Eventually, the learner we will consider does not update its regression vector at every round, but rather periodically, so that individuals can be treated in batches.

Notice that any discovery of meaningful variables that occurs in our model is a result of the *interaction* between the online learner and the strategic individuals. On the one hand, online learning with no strategic response has no ability to distinguish non-meaningful variables from meaningful ones when the two are correlated. On the other hand, if strategic individuals faced with a static scoring algorithm tried to maximize their scores by investing in a non-meaningful feature, the resulting information would be insufficient for an observer to draw conclusions about whether other features are meaningful or not.

For example, historical data might show that both a student's grades in high school and the make of car his parents drive to the university visit day are predictive of success in university. Suppose, for simplicity, that success in high school is causally related to success in university, but that make of parents' car is not, and is merely a *proxy* for other features that control one's chances of success in college. If the university admissions process put large weight on high school grades, that would incentivize students to invest effort in performing well in high school, which would also observably pay off in university, which would reinforce the emphasis on high school grades. If the admissions process put large weight on the make of car in which students arrive to the visit day, that would incentivize renting fancy cars for visits. However, this would result in a different distribution over the observed student variables, and on this modified distribution the correlation between cars and university success would be *weakened*, and therefore the admissions formula would not perform well. In future years, the university would naturally correct the formula to deemphasize cars.

It is important to note that our work operates under a simplifying assumption with regards to the underlying structure of the problem (introduced in Section 3). Adding an assumption of this kind is necessary since in the general case recovering the exact model structure is hard. Our work thus aims to bring attention to a natural mechanism, based on re-training, for exposing meaningful variables, that we believe is worthy of further attention.

## 2 Related Work

Much of the work on learning assumes that an individual's data is a fixed input that is independent of the algorithm used by the decision-maker. In practice, however, individuals may try to adapt to the model in place in order to improve their outcomes. A recent line of work studies such strategic behavior in classification settings. Part of this line of work concerns itself

with the negative consequences of strategic behavior, when individuals aim to game the model in place; for example, individuals may manipulate their data or features (often at a cost) in an effort to obtain positive qualification outcomes or otherwise manipulate an algorithm's output [6, 19, 7, 1, 14, 12, 2, 11, 8, 4, 3] or even to protect their privacy [9, 5]. The goal in these results is to provide algorithms whose outputs are robust to such gaming. [17] and [13] focus on the social impact of robust classification, and show that i) robust classifiers come at a social cost (by forcing even qualified individuals to invest in costly feature manipulations in order to be classified positively) and ii) disparate abilities to game the model inevitably lead to unfair outcomes.

Another part of this line of work instead sees strategic manipulation as possibly positive, when the classifier incentivizes individuals to invest in true improvements to their features—e.g., a student may decide to study and actually improve his knowledge in order to raise his test score. [15], [23], [21], [22] and [10] study how to incentivize agents to invest effort in modifying meaningful features that improve their labels. Much of this line of work assumes that the decision-maker already understands which features are meaningful and affect agents' labels or outcomes, and which do not.

In contrast, we consider a setting where the decision-maker does not initially know which features affect agents' labels, and aims to leverage the agents' strategic behavior to expose what features these are. Most closely related to this paper is the work of [16], as well as the concurrent works of [18] and [20]. [16] formalize the distinction between gaming and actual improvements by drawing a connection to causality and introducing causal graphs that model the effects of the features and target variables on each other. They show that in such settings, the decision-maker should incentivize actual improvements rather than gaming, and that designing good incentives that push agents to improve is at least as hard as causal inference. [20] study the sample complexity of learning a linear regression model so as to either i) maximize the accuracy of the predictions, ii) maximize the agents' self-improvements, or iii) recover the causality structure of their problem. [18] show how re-training can lead to stable and optimal outcomes when the learner's model affects the distribution of agent features and labels; while our paper considers a similar re-training framework, our assumptions differ from those of [18].

## 3 Model

We consider a linear regression setting where the learner estimates the regression parameters based on strategically manipulated data from a sequence of agents over rounds. There is a true latent regression parameter $\beta^* \in [-1,1]^d$ that *generates* an agent's label as a function of his feature vector. That is, for any agent with feature vector $x \in [-1,1]^d$, the real-valued label $y$ is obtained via $y = \beta^{*\top} x + \varepsilon$, where $\varepsilon$ is a noise random variable with $|\varepsilon| \leq \sigma$, and $\mathbb{E}[\varepsilon \mid x] = 0$. We also refer to an individual's features as variables. There is a distribution over the *unmodified* features $x$ in $[-1,1]^d$; we let $\mu$ be the mean and $\Sigma$ be the covariance matrix of this distribution; we note that the distribution of unmodified features may be degenerate, i.e., $\Sigma$ may not be full-rank. For example, this can happen in settings in which the non-meaningful features are merely proxies for the meaningful features (i.e., those that really control the label); in that case, one may imagine that the non-meaningful features are (possibly randomized) functions of the meaningful features, leading in particular to low-rank observations when few features are meaningful. Throughout the paper, we set $\mu = 0$.[2]

The agents and the learner interact in an online fashion. At time $t$, the learner first posts a regression estimate $\hat{\beta}^t \in \mathbb{R}^d$, then an agent (indexed by $t$) arrives with their unmodified feature vector $x_t$. Agent $t$ modifies the feature $x_t$ into $\bar{x}_t$ in response to $\hat{\beta}^t$, in order to improve their assigned score $\langle \hat{\beta}^t, \bar{x}_t \rangle$. Finally, the learner observes the agent's realized label after feature modification, given by $\bar{y}_t = \beta^{*\intercal} \bar{x}_t + \varepsilon_t$.

**Meaningful vs non-meaningful features.** When an agent modifies a feature $k$, this may also affect the agent's true label. We divide the coordinates of any feature vector $x$ into *meaningful* and *non-meaningful* features; meaningful features inform and control an agent's label, while non-meaningful features are those that can be manipulated without directly affecting an agent's label. (One can think, intuitively, of the meaningful features as causal, and the non-meaningful features as non-causal, but the language of causality is typically reserved for more complex settings than ours.) Formally, for any $k \in [d]$, feature $k$ is meaningful if and only if the coordinate $\beta^*(k) \neq 0$, and non-meaningful if and only if $\beta^*(k) = 0$. An agent $t$ can modify his true label by modifying meaningful features. As such, note that $\beta^*$ captures the underlying model structure of our problem. The magnitude of each feature in $\beta^*$ captures the extent to which said

[2] This can be done whenever the learner can estimate the mean feature vector, since the learner can then center the features. The learner could estimate the mean by using unlabeled historical data; for example, she could collect data during a period when the algorithm does not make any decision on the agents, thus they would have no incentive to modify their features.

feature is meaningful and affects the agents' labels.

We remark that strategic agents—that best-respond to the learner's model to improve their regression outcomes—may at times have incentives to manipulate a feature $k$ such that $\beta^*(k) = 0$; this can happen when the learner sets $\hat{\beta}(k) \neq 0$. In such cases, agents can improve their regression outcomes *without* improving their true label, which we refer to as *gaming*. When agents modify a feature $k$ that aligns with the true model, we refer to such a modification as an *improvement*.

**Agents' responses.** Agents are *strategic*: they modify their features so as to maximize their own regression outcome;[3] modifications are costly and agents are budgeted. We assume agent $t$ incurs a linear cost $c_t(\Delta_t) = \sum_{k=1}^{d} c_t(k)\,|\Delta_t(k)|$ to change his features by $\Delta_t$, and has a total budget of $B_t$ to modify his features. $\left(\{c_t(k)\}_{k\in[d]}, B_t\right)$'s are drawn i.i.d. from a distribution $\mathcal{C}$ that is unknown to the learner. We assume $\mathcal{C}$ has discrete support $\{\left(c^1, B^1\right), \ldots, \left(c^l, B^l\right)\}$, and we denote by $\pi^i$ the probability that $(c_t, B_t) = \left(c^i, B^i\right)$. We assume $c^i(k) > 0,\ B^i > 0$ for all $i \in [l],\ k \in [d]$; that is, every agent can modify his features, but no feature can be modified for free.[4] When facing regression parameters $\hat{\beta}$, agent $t$ solves

$$M(\hat{\beta}, c_t, B_t) = \underset{\Delta_t}{\operatorname{argmax}} \quad \hat{\beta}^\top (x_t + \Delta_t)$$
$$\text{s.t.} \sum_{k=1}^{d} c_t(k)\,|\Delta_t(k)| \leq B_t;$$

That is, agent $t$ strategically aims to maximize his predicted outcome given a budget of $B$ for modifying his features, when facing model $\hat{\beta}$. The solution of the above program does not depend on $x_t$, only on $\hat{\beta}$ and $(c_t, B_t)$, and is given by

$$\Delta_t = \sum_{k=1}^{d} sgn\left(\hat{\beta}(k)\right) \cdot \ldots$$
$$\ldots \cdot \mathbb{1}\left\{k = \underset{j}{\operatorname{argmax}} \left|\hat{\beta}(j)\right| / c_t(j)\right\} \frac{B_t}{c_t(k)},$$

up to tie-breaking; when several features maximize $|\hat{\beta}(j)|/c_t(j)$, the agent modifies a single one of these features. We call $D_\tau$ the set of features that have been modified by at least one agent $t \in [\tau]$.

**Remark 3.1.** *We make the linearity assumption on the cost functions for simplicity. Our results extend to a more general class of cost functions that do not induce modifications wherein several features are modified in a perfectly correlated fashion.*

*The key technical insight we need is that the manipulations are full-rank in the subspace defined by the features that have been manipulated so far, defined as $\mathcal{V}_{\tau(E)}$ in the paper. Very strong feature correlations (which may also be thought of possible "directions" for modification) imply a very small minimum eigenvalue of the observation matrix, making recovery harder and increasing sample complexity. This is unavoidable: the more features are correlated, the harder they are to distinguish information-theoretically; if two features were perfectly correlated, it would be impossible to know which one affected the label.*

*In Theorem 4.1, we encode this correlation between modification across features in a parameter we call $\lambda$. As feature modifications become more and more correlated, the value of $\lambda$ becomes smaller and our recovery guarantees weaken.*

**Natural learner dynamics: batch least-squares regression.** Our goal here is to identify simple, natural learning dynamics that expose meaningful variables. Note that a simple way for the learner to expose and leverage meaningful variables is to use an explore-first then exploit type of algorithm: initially, the learner can post a model that focuses on a single feature at a time to observe how changing this feature affects the distribution of agents labels. After sequentially exploring each feature, the learner obtains an accurate estimate of $\beta^*$ that she can deploy for the remainder of the time horizon. However, one may want to avoid such an approach that artificially separates features in practice: posting models that ignore most of an agent's attribute for the sake of learning may not be desirable in real life. A bank may not want to offer loans "blindly" and willingly ignore most of a customer's data when making lending decisions just for the purpose of learning which features are predictive of an agent's ability to repay loans. Instead, in this paper, we focus on algorithms based on *re-training*: i.e., periodically, the learner updates her model based on the data she has observed so far, so as to keep it consistent with the history of agent behavior. A bank may be willing to periodically update their loan decision rule in order to keep up with new, unexpected agent behavior. While re-training leads to more natural dy-

[3] Importantly, our agents' goal *is not* to cooperate with the learner. Agents are self-interested and aim to maximize their own regression outcomes; they do not actively seek to help the learner improve the accuracy of her model. The agents prefer when the learner emphasizes features that are easier to manipulate, even if said features are non-meaningful. These incentives may be ill-aligned with the learner's goal of optimizing predictive power and recovering model structure, which requires putting more weight on meaningful features.

[4] In our model, modifying a feature affects only that feature and the label, but does not affect the values of any other features. We leave exploration of more complex models of feature intervention to future work.

namics than a "naive" explore-then-exploit approach, it comes with new technical challenges. In particular, periodic re-training leads to *adaptivity*: indeed, as the model posted in the current period depend on past data, and the agents' strategic behavior depends on the model in place, the observed *modified* data in each period depends on the data in all previous periods. In turn, we cannot treat data points as independent across periods.

The dynamics we consider are formally given in Algorithm 1. It is possible that more sophisticated learning algorithms could yield better guarantees with respect to regret and recovery; the focus of this paper is on simple and natural dynamics rather than optimal ones.

When the learner updates her regression parameters, say at time $\tau$, she does so based on the agent data observed up until time $\tau$. We model the learner as picking $\hat{\beta}$ from the set $LSE(\tau)$ of solutions to the least-square regression problem run on the agents' data up until time $\tau$, formally defined as

$$LSE(\tau) = \underset{\beta}{\operatorname{argmin}} \sum_{t=1}^{\tau} \left(\bar{x}_t^\top \beta - \bar{y}_t\right)^2.$$

We introduce notation that will be useful for regression analysis. We let $\bar{X}_\tau \in \mathbb{R}_{\tau \times d}$ be the matrix of (modified) observations up until time $\tau$. Each row corresponds to an agent $t \in [\tau]$, and agent $t$'s row is given by $\bar{x}_t^\top$. Similarly, let $\bar{Y}_\tau = (\bar{y}_t)_{t\in[\tau]}^\top \in \mathbb{R}^{\tau \times 1}$. We can rewrite, for any $\tau$,

$$LSE(\tau) = \underset{\beta}{\operatorname{argmin}} \left(\bar{X}_\tau \beta - \bar{Y}_\tau\right)^\top \left(\bar{X}_\tau \beta - \bar{Y}_\tau\right). \quad (1)$$

**Agents are grouped in epochs.** The time horizon $T$ is divided into epochs of size $n$, where $n$ is chosen by the learner. At the start of every epoch $E$, the learner updates the posted regression parameter vector as a function of the history of $\bar{x}_t, \bar{y}_t$ up until epoch $E$. We let $\tau(E) = En$ denote the last time step of epoch $E$. $D_{\tau(E)}$ denotes the set of features that have been modified by at least one agent by the end of epoch $E$.

**Algorithm 1:** Online Regression with Epoch-Based Strategic modification (Epoch size $n$)

Learner picks (any) initial $\hat{\beta}_0$.
**for** *every epoch* $E \in \mathbb{N}$ **do**
  **for** $t \in \{(E-1)n+1, \ldots, En\}$ **do**
    Agent $t$ reports $\bar{x}_t \in M(\hat{\beta}_{E-1}, c_t, B_t)$.
    Learner observes $\bar{y}_t = \beta^{*\top} \bar{x}_t + \varepsilon_t$.
  **end**
  Learner picks $\hat{\beta}_E \in LSE\left(\tau(E)\right)$.
**end**

**Examples** We first illustrate why unmodified observations are insufficient for *any* algorithm to distinguish meaningful from non-meaningful features. Consider a setting where non-meaningful features, as merely proxies for the meaningful features, are in fact convex combinations of these meaningful features in the underlying (unmodified) distribution. Absent additional information, a learner would be faced with degenerate sets of observations that have rank strictly less than $d$, which can make accurate recovery of the model structure impossible:

**Example 3.2.** *Suppose $d = 2$, $\beta^* = (1, 0)$. Suppose feature 1 is meaningful and feature 2 is non-meaningful and is correlated with 1: the distribution of unmodified features is such that for any feature vector $x$, feature 2 is identical to feature 1 as $x(2) = x(1)$. Then, any regression parameter of the form $\beta(\alpha) = (\alpha, 1-\alpha)$ for $\alpha \in \mathbb{R}$ assigns agents the same score as $\beta^*$. Indeed,*

$$\beta^{*\top} x = x(1) = \alpha x(1) + (1-\alpha)x(2) = \beta(\alpha)^\top x.$$

*In turn, in the absence of additional information other than the observed features and labels, $\beta^*$ is indistinguishable from any $\beta(\alpha)$, many of which recover the model structure poorly (e.g., consider any $\alpha$ bounded away from 1).*

At this point, a reader may wonder why it is important in Example 3.2 to recover the true model $\beta^*$, rather than simply *any* vector $\beta$ that is consistent with all the data observed so far. A major reason to do so is because only the true model $\beta^*$ can guarantee robustness in response to agent modifications, and accurately predict labels *after* agents have changed their features. This is illustrated in Example 3.3 below:

**Example 3.3.** *Consider the setting of Example 3.2, and imagine agents have much lower cost for manipulating feature 2 than feature 1. Then, posting a regression parameter vector of the form $(\alpha, 1-\alpha)$ where $\alpha$ is small enough may lead agents to modify the second, non-meaningful feature. When facing such a modification of the form $\Delta = (0, \Delta(2))$, $(\alpha, 1-\alpha)$ predicts label*

$$\alpha x(1) + (1-\alpha)\left(x(2) + \Delta(2)\right) = x(1) + (1-\alpha)\Delta(2),$$

*for an agent with $x(1) = x(2)$, while the true label is given by $\beta^{*\top}(x + \Delta) = x(1)$. In turn, the predicted and true labels are different for any $\alpha \neq 1$.*

We next illustrate that strategic agent modifications may aid in recovery of meaningful features, but only for those features that individuals actually invest in changing:

**Example 3.4.** *Consider a setting where $d = 3$, feature 1 is meaningful, and features 2 and 3 are non-meaningful and are correlated with feature 1 as follows: for any feature vector $x$, $x(2), x(3) = x(1)$. Let*

$\beta^* = (1, 0, 0)$. *Consider a situation in which the labels are noiseless (i.e., $\varepsilon = 0$ almost surely). Suppose that agents only modify their meaningful feature by a (possibly random) amount* $\Delta(1)$.

*Note that the difference (in absolute value) between the score obtained by applying a given regression parameter $\hat{\beta}$ and the score obtained by applying $\beta^*$ to feature vector $x$ is given by*

$$\begin{aligned}&\left|\hat{\beta}^\top x - \beta^{*\top} x\right| \\ &= \left|\hat{\beta}(1)\left(x(1) + \Delta(1)\right) + \hat{\beta}(2)x(2) + \ldots \right. \\ &\qquad\qquad \left. \ldots + \hat{\beta}(3)x(3) - x(1) - \Delta(1)\right| \\ &= \left|\left(\hat{\beta}(1) + \hat{\beta}(2) + \hat{\beta}(3) - 1\right) x(1) + \left(\hat{\beta}(1) - 1\right)\Delta(1)\right|.\end{aligned}$$

*In particular, for appropriate distributions of $x$ and $\Delta(1)$, the predictions of $\hat{\beta}$ and $\beta^*$ coincide if only if $\hat{\beta}(1) = 1$ and $\hat{\beta}(2) = -\hat{\beta}(3)$. As such, the learner learns after enough observations that necessarily, $\beta^*(1) = 1$. However, any regression parameter vector with $\hat{\beta}(1) = 1$, $\hat{\beta}(2) + \hat{\beta}(3) = 0$ is indistinguishable from $\beta^*$, and accurate recovery of $\beta^*(2)$ and $\beta^*(3)$ is impossible.*

Note that even in the noiseless setting of Example 3.4, only the feature that has been modified can be recovered accurately. In more complex settings where the true labels are noisy, one should not hope to recover every feature well, but rather only those that have been modified sufficiently many times.

## 4 Recovery Guarantees for Modified Features

In this section, we focus on characterizing the recovery guarantees (with respect to the $\ell_2$-norm) of Algorithm 1 at time $\tau(E) = En$ for *any* epoch $E$, with respect to the features that have been modified up until $\tau(E)$ (that is, in epochs 1 to $E$). We leave discussion of how the dynamics shape the set $D_{\tau(E)}$ of modified features to Section 5.

The main result of this section guarantees the accuracy of the $\hat{\beta}_E$ that the learning process converges to in its interaction with a sequence of strategic agents. The accuracy of the $\hat{\beta}_E$ that is recovered for a particular feature naturally depends on the number of epochs in which that feature is modified by the agents. For a feature that is never modified, we have no ability to distinguish whether it is meaningful or not. Recovery improves as the number of observations of the modified variable increases.

Formally, our recovery guarantee is given by the following theorem:

**Theorem 4.1** ($\ell_2$ Recovery Guarantee for Modified Features)**.** *Pick any epoch $E$. With probability at least $1 - \delta$, for $n \geq \frac{\kappa d^2}{\lambda}\sqrt{\tau(E)\log(12d/\delta)}$,*

$$\sqrt{\sum_{k \in D_{\tau(E)}} \left(\hat{\beta}_E(k) - \beta^*(k)\right)^2} \leq \frac{K\sqrt{d\tau(E)\log(4d/\delta)}}{\lambda n},$$

*where $K$, $\kappa$, $\lambda$ are instance-specific constants that only depend on $\sigma$, $\mathcal{C}$, $\Sigma$, such that $\lambda > 0$.*

When the epoch size is chosen so that $n = \Omega\left(\tau(E)^\alpha\right)$ for $\alpha > 1/2$, our recovery guarantee improves as $\tau(E)$ becomes larger.

Now, let us fix $\tau(E) = T$ as the time horizon, and study how the relationship between $E$ and $n$ at fixed $\tau(E)$ affects the recovery guarantees. When $n = \Theta(\tau(E))$ (equivalently, $E = \Theta(1)$, and agents are grouped in a small, constant number of epochs), our bound becomes $O(1/\sqrt{\tau(E)})$; this matches the well-known recovery guarantees of least square regression for a single batch of $\tau(E)$ i.i.d observations drawn from a non-degenerate distribution of features. When the epoch size $n$ is sub-linear in $\tau(E)$ (i.e., $E \gg 1$, and agents are grouped in more numerous but smaller epochs),the accuracy guarantee degrades to $O(\sqrt{\tau(E)}/n)$, where $\sqrt{\tau(E)}/n \gg \frac{1}{\sqrt{\tau(E)}}$. This is because some features may be modified only in a small number of epochs,[5] that is, $\Theta(n)$ times, and the number of times such features are modified drives how accurately they can be recovered.

*Proof sketch for Theorem 4.1.* Full proof in Appendix A. We focus on the subspace $\mathcal{V}_{\tau(E)}$ of $\mathbb{R}^d$ spanned by the observed features $\bar{x}_1, \ldots, \bar{x}_{\tau(E)}$, and for any $z \in \mathbb{R}^d$, we denote by $z(\mathcal{V}_{\tau(E)})$ the projection of of $z$ onto $\mathcal{V}_{\tau(E)}$. First, we show via concentration that in this subspace, the mean-square error is strongly convex, with parameter $\Theta(n)$ (see Claim A.6). This strong convexity parameter is controlled by the smallest eigenvalue of $\bar{X}_{\tau(E)}^\top \bar{X}_{\tau(E)}$ over subspace $\mathcal{V}_{\tau(E)}$. Formally, we lower bound this eigenvalue and show that with probability at least $1 - \delta/2$, for $n$ large enough,

$$\begin{aligned}&\left(\hat{\beta}_E(\mathcal{V}_{\tau(E)}) - \beta^*(\mathcal{V}_{\tau(E)})\right)^\top \bar{X}_{\tau(E)}^\top \bar{X}_{\tau(E)} \cdot \ldots \\ &\qquad \ldots \cdot \left(\hat{\beta}_E(\mathcal{V}_{\tau(E)}) - \beta^*(\mathcal{V}_{\tau(E)})\right) \geq \frac{\lambda n}{4}. \end{aligned} \tag{2}$$

[5] In particular, as we will see, we expect correlated, non-meaningful features to only be modified in a small number of epochs: once a non-meaningful feature $k$ has been modified in a few epochs, it is accurately recovered. In further periods $E$, the learner sets $\hat{\beta}_E(k)$ close to 0. This disincentivizes further modifications of feature $k$.

Second, we bound the effect of the noise $\varepsilon$ on the mean-squared error by $O(\sqrt{\tau(E)})$ in Lemma A.3, once again via concentration. Formally, we abuse notation and let $\varepsilon_{\tau(E)} \triangleq (\varepsilon_t)^\top_{t\in[\tau(E)]}$, and show that with probability at least $1-\delta/2$,

$$\left(\hat{\beta}_E(\mathcal{V}_{\tau(E)}) - \beta^*(\mathcal{V}_{\tau(E)})\right)^\top \bar{X}^\top_{\tau(E)}\varepsilon_{\tau(E)} \leq \left\|\hat{\beta}_E(\mathcal{V}_{\tau(E)}) - \beta^*(\mathcal{V}_{\tau(E)})\right\|_2 \cdot K\sqrt{d\tau(E)\log(4d/\delta)}. \tag{3}$$

Finally, we obtain the result via Lemma A.2, that states that taking the first-order conditions on the mean-squared error yields

$$\bar{X}^\top_{\tau(E)}\bar{X}_{\tau(E)}\left(\hat{\beta}_E(\mathcal{V}_{\tau(E)}) - \beta^*(\mathcal{V}_{\tau(E)})\right) = \bar{X}^\top_{\tau(E)}\varepsilon_{\tau(E)},$$

which can be combined with Equations (2) and (3) to show our bound with respect to sub-space $\mathcal{V}_{\tau(E)}$. In turn, as $D_{\tau(E)}$ defines a sub-space of $\mathcal{V}_{\tau(E)}$, our accuracy bound applies to $D_{\tau(E)}$. □

**Remark 4.2.** *Theorem 4.1 is not a direct consequence of the classical recovery guarantees of least-square regression, as they assume $\bar{X}^\top_{\tau(E)}\bar{X}_{\tau(E)}$ has full rank $d$. We deal with degenerate distributions over modified features, that can arise in our setting as per Examples 3.2 and 3.4.*

## 5 Exploration via Least Squares Tie-Breaking

In this section, we show that a natural tie-breaking rule among the set of least squares incentivizes agents' modification of a diverse set of variables over time.

Recall we are solving the least-square problem $LSE(\tau(E))$ given in Equation (1) for all epochs $E$. When $\bar{X}^\top_{\tau(E)}\bar{X}_{\tau(E)}$ is invertible, this has a single solution. However, in our setting, it may be the case that $\bar{X}^\top_{\tau(E)}\bar{X}_{\tau(E)}$ is rank-deficient (see Examples 3.2, 3.4). In this case, the least-square problem admits a continuum of solutions. This gives rise to the question of which solutions are preferable in our setting, and how to break ties between several solutions.

The learner's choice of regression parameters in each epoch affects the distribution of feature modifications in subsequent epochs. As the recovery guarantee of Theorem 4.1 only applies to features that have been modified, we would like our tie-breaking rule to regularly incentivize agents to modify new features. We first show that a natural, commonly used tie-breaking rule—picking the minimum norm solution to the least-square problem—may fail to do so:

**Example 5.1.** *Consider a setting with $d = 2$, $\beta^* = (1, 2)$ and noiseless labels, i.e., $\varepsilon_t = 0$ always. Suppose that with probability 1, every agent $t$ has features $x_t = (0, 0)$, budget $B_t = 1$, and costs $c_t(1) = c_t(2) = 1$ to modify each feature. We let the tie-breaking pick the solution with the least $\ell_2$ norm among all solutions to the least-square problem.*

*Pick any initial regression parameter $\hat{\beta}_0$ with $\hat{\beta}_0(1) > \hat{\beta}_0(2)$. For every agent $t$ in epoch 1, $t$ picks modification vector $\Delta_t = (1, 0)$. This induces observations $\bar{x}_t = (1, 0)$, $\bar{y}_t = 1$. The set of least-square solutions (with error exactly 0) in epoch 1 is then given by $\{(1, \beta_2) : \forall \beta_2 \in \mathbb{R}\}$, and the minimum-norm solution chosen at the end of epoch 1 is $\hat{\beta}_1 = (1, 0)$. This solution incentivizes agents to set $\Delta_t = (1, 0)$, and Algorithm 1 gets stuck in a loop where every agent $t$ reports $\bar{x}_t = (1, 0)$, and the algorithm posts regression parameter vector $\hat{\beta}_E = (1, 0)$ in response, in every epoch $E$. The second feature is never modified by any agent, and is not recovered accurately.*

Example 5.1 highlights that a wrong choice of tie-breaking rule can lead Algorithm 1 to explore the same features over and over again. In response, we propose the following tie-breaking rule, described in Algorithm 2: Intuitively, at the end of epoch $E$, our tie-breaking rule picks a solution in $LSE(\tau(E))$ with large norm. This ensures the existence of a feature $k \notin D_{\tau(E)}$ that has not yet been modified up until time $\tau(E)$, and that is assigned a large weight by our least-square solution. In turn, this feature is more likely to be modified in future epochs.

**Algorithm 2:** Tie-Breaking Scheme at Time $\tau(E)$.

**Input:** Epoch $E$, observations $(\bar{x}_1, \bar{y}_1), \ldots, (\bar{x}_{\tau(E)}, \bar{y}_{\tau(E)})$, parameter $\alpha$

Let $\mathcal{U}_{\tau(E)} = \text{span}\left(\bar{x}_1, \ldots, \bar{x}_{\tau(E)}\right)$.

**if** *rank* $\left(\mathcal{U}_{\tau(E)}\right) < d$ **then**

  Find an orthonormal basis $B^\perp_{\tau(E)}$ for $\mathcal{U}^\perp_{\tau(E)}$.

  Set $v = \sum_{b\in B^\perp_{\tau(E)}} b \neq 0$, renormalize $v := \frac{v}{\|v\|_2}$.

  Pick $\beta_E$ a vector in $LSE(\tau(E))$ with minimal norm.

  Set $\hat{\beta}_E = \beta_E + \alpha v$.

**else**

  Set $\hat{\beta}_E$ be the unique element in $LSE(\tau(E))$.

**end**

**Output:** $\hat{\beta}_E$.

Our main result in this section shows that the tie-breaking rule of Algorithm 2 eventually incentivizes the agents to modify all $d$ features, allowing for accurate recovery of $\beta^*$ in its entirety. The intuition behind our algorithm is to choose a tie-breaking rule that puts

enough weight on directions that have not yet been explored, incentivizing agents to explore them.

**Theorem 5.2** (Recovery Guarantee with Tie-Breaking Scheme (Algorithm 2))**.** *Suppose the epoch size satisfies* $n \geq \frac{\kappa d^2}{\lambda}\sqrt{2T\log(24d/\delta)}$*, and take* $\alpha$ *to be*

$$\alpha \geq \gamma\left(\sqrt{d} + \frac{Kd\sqrt{2T\log(8d/\delta)}}{\lambda n}\right),$$

*where* $\gamma$*,* $K$*,* $\kappa$*,* $\lambda$ *are instance-specific constants that only depend on* $\sigma$*,* $\mathcal{C}$*,* $\Sigma$*, and* $\lambda > 0$*. If* $T \geq dn$*, we have with probability at least* $1-\delta$ *that at the end of the last epoch* $T/n$*,*

$$\left\|\hat{\beta}_{T/n} - \beta^*\right\|_2 \leq \frac{K\sqrt{2dT\log(8d/\delta)}}{\lambda n},$$

*under the tie-breaking rule of Algorithm 2 .*

**Remark 5.3.** *The bound in Theorem 5.2 provides guidance for selecting the epoch length, so as to ensure optimal recovery guarantees. Under the natural assumption that* $T >> d$*, the optimal recovery rate is achieved when roughly* $n = \Theta(T/d)$*. This results in an* $O(d\sqrt{(d\log d)/T})$ *upper bound on the* $\ell_2$ *distance between the recovered regression parameters and* $\beta^*$*.*

*Proof sketch of Theorem 5.2.* Full proof in Appendix B. For $\alpha$ arbitrarily large, the norm of $\hat{\beta}$ becomes arbitrarily large. Because at the end of epoch $E$, $\hat{\beta}_E$ guarantees accurate recovery of all features modified up until time $En$, it must be that $\hat{\beta}_E(k)$ is arbitrarily large for some feature $k$ that has not yet been modified. In turn, this feature is modified in epoch $E+1$. After $d$ epochs, and in particular for $T \geq dn$, this leads to $D_T = [d]$. The recovery guarantee of Theorem 4.1 then applies to all features. □

## 6 Conclusion

This work takes a first step towards illuminating a phenomenon we believe is both surprising and worthy of further study: strategic agents may in fact help a learner in better understanding the underlying structure of a classification problem. As an immediate implication, the recovery guarantees we have proven provide the learner with knowledge regarding how to choose good incentives, laying the ground for individual improvement, rather than gaming. In future work, it would be natural to explore this interaction in richer and more complex settings.

### Acknowledgements

Part of this work was done while the authors were visiting the Simons Institute for the Theory of Computing. The work of Yahav Bechavod and Katrina Ligett was supported in part by Israel Science Foundation (ISF) grants #1044/16 and 2861/20, the United States Air Force and DARPA under contracts FA8750-16-C-0022 and FA8750-19-2-0222, and the Federmann Cyber Security Center in conjunction with the Israel national cyber directorate. Yahav Bechavod was also supported in part by the Apple Scholars in AI/ML PhD Fellowship. Katrina Ligett was also funded in part by in part by a grant from Georgetown University and Simons Foundation Collaboration 733792. Zhiwei Steven Wu was supported in part by the NSF FAI Award #1939606, a Google Faculty Research Award, a J.P. Morgan Faculty Award, a Facebook Research Award, and a Mozilla Research Grant. Juba Ziani was supported in part by the Inaugural PIMCO Graduate Fellowship at Caltech, the National Science Foundation through grant CNS-1518941, as well as the Warren Center for Network and Data Sciences at the University of Pennsylvania. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force and DARPA. We thank Mohammad Fereydounian and Aaron Roth for useful discussions.

### References

[1] Michael Brückner, Christian Kanzow, and Tobias Scheffer. Static prediction games for adversarial learning problems. *Journal of Machine Learning Research*, 13(Sep):2617–2654, 2012.

[2] Yang Cai, Constantinos Daskalakis, and Christos H. Papadimitriou. Optimum statistical estimation with strategic data sources. In *COLT*, 2015.

[3] Yiling Chen, Yang Liu, and Chara Podimata. Grinding the space: Learning to classify against strategic agents. *arXiv preprint arXiv:1911.04004*, 2019.

[4] Yiling Chen, Chara Podimata, Ariel D Procaccia, and Nisarg Shah. Strategyproof linear regression in high dimensions. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 9–26, 2018.

[5] Rachel Cummings, Stratis Ioannidis, and Katrina Ligett. Truthful linear regression. In *COLT*, 2015.

[6] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, 2004.

[7] Ofer Dekel, Felix Fischer, and Ariel D. Procaccia. Incentive compatible regression learning. *Journal of Computer and System Sciences*, 76(8):759 – 777, 2010.

[8] Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 55–70, 2018.

[9] Arpita Ghosh, Katrina Ligett, Aaron Roth, and Grant Schoenebeck. Buying private data without verification. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, pages 931–948, 2014.

[10] Nika Haghtalab, Nicole Immorlica, Brendan Lucier, and Jack Wang. Maximizing welfare with incentive-aware evaluation mechanisms. Technical report, working paper, 2020.

[11] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122, 2016.

[12] Thibaut Horel, Stratis Ioannidis, and S. Muthukrishnan. Budget feasible mechanisms for experimental design. In *LATIN 2014: Theoretical Informatics*, Lecture Notes in Computer Science, pages 719–730, 2014.

[13] Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 259–268, 2019.

[14] Stratis Ioannidis and Patrick Loiseau. Linear regression as a non-cooperative game. In *Web and Internet Economics*, pages 277–290, 2013.

[15] Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? In *Proceedings of the FAT**, pages 825–844, 2019.

[16] John Miller, Smitha Milli, and Moritz Hardt. Strategic adaptation to classifiers: A causal perspective. *arXiv preprint arXiv:1910.10362*, 2019.

[17] Smitha Milli, John Miller, Anca D Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 230–239, 2019.

[18] Juan C Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. *arXiv preprint arXiv:2002.06673*, 2020.

[19] Javier Perote and Juan Perote-Pena. Strategy-proof estimators for simple regression. In *Mathematical Social Sciences 47*, pages 153–176, 2004.

[20] Yonadav Shavit, Benjamin Edelman, and Brian Axelrod. Learning from strategic agents: Accuracy, improvement, and causality. *arXiv preprint arXiv:2002.10066*, 2020.

[21] Behzad Tabibian, Stratis Tsirtsis, Moein Khajehnejad, Adish Singla, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Optimal decision making under strategic behavior. *arXiv preprint arXiv:1905.09239*, 2019.

[22] Stratis Tsirtsis and Manuel Gomez-Rodriguez. Decisions, counterfactual explanations and strategic behavior. *arXiv preprint arXiv:2002.04333*, 2020.

[23] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 10–19, 2019.