Low-Cost Learning via Active Data Procurement

JACOB ABERNETHY, University of Michigan
YILING CHEN, Harvard University
CHIEN-JU HO, UCLA
BO WAGGONER, Harvard University

We design mechanisms for online procurement of data held by strategic agents for machine learning tasks. We study a model in which agents cannot fabricate data, but may lie about their cost of furnishing their data. The challenge is to use past data to actively price future data in order to obtain learning guarantees, even when agents' costs can depend arbitrarily on the data itself. We show how to convert a large class of no-regret algorithms into online posted-price and learning mechanisms. Our results parallel classic sample complexity guarantees, but with the key resource constraint being money rather than quantity of data available. With a budget constraint B, we give robust risk (predictive error) bounds on the order of $1/\sqrt{B}$. In many cases our guarantees are significantly better due to an active-learning approach that leverages correlations between costs and data.

Our algorithms and analysis go through a model of no-regret learning with T arriving pairs (cost, data) and a budget constraint of B, coupled with the "online to batch conversion". Our regret bounds for this model are on the order of $T/\sqrt{B}$ and we give lower bounds on the same order.

Categories and Subject Descriptors: J.4 [**Social and Behavioral Sciences**]: Economics; I.2.6 [**Artificial Intelligence**]: Learning

Additional Key Words and Phrases: machine learning; online learning; mechanisms; data procurement

1. INTRODUCTION

The rising interest in the field of Machine Learning (ML) has been strongly driven by the potential to generate economic value. Firms seeking revenue optimizations can gather abundant data at low cost, apply a set of inexpensive algorithmic tools, and produce high-accuracy predictors that can massively improve future decision making. The extent of the potential value that can be created by leveraging data for prediction is apparent in the multi-million dollar competition bounties offered by companies like Netflix and the Heritage Health Foundation, but perhaps even more so in the aggressive hiring of many ML experts by companies like Google and Facebook.

Much of the theoretical results in ML aim to measure, at least implicitly, the *economic efficiency* of learning problems. For example, in certain settings we have a reasonably thorough understanding of *sample complexity* [Anthony and Bartlett 2009] which gives us the precise tradeoff between n, the quantity of data at our disposal, and the error or loss rate we want to achieve. Reducing error is always beneficial, of course, but must be weighed against the marginal cost of increasing n.

The measures of efficiency in ML have broadened in recent years, in particular because gathering data is typically orders of magnitude cheaper than labeling it. This has led to the emergence of the *active learning* paradigm [Balcan et al. 2006, 2010;

Addresses: jabernet@umich.edu, yiling@seas.harvard.edu, cjho@ucla.edu, bwaggoner@fas.harvard.edu.

EC'15, June 15–19, 2015, Portland, OR, USA. ACM 978-1-4503-3410-5/15/06 ...$15.00.

http://dx.doi.org/10.1145/2764468.2764519

Beygelzimer et al. 2009; Hanneke 2009; Settles 2011]. Here, we imagine an interface between the learner and the label provider, where the learner may make label queries on data points in an online fashion. By sequentially choosing which data to label, the learner can greatly reduce the number of labels required to learn [Hanneke 2009].

A problem that has received little attention in the learning theory literature is the *monetary efficiency* of learning when data have differing costs. Indeed, real-world prediction tasks often require obtaining examples held by self-interested, strategic agents; these agents must be incentivized to provide the data they hold, and they have heterogeneous costs for doing so.

In this vein, the present paper seeks to address the following question:

> In a world where data is held by self-interested agents with heterogeneous costs for providing it, and in particular when these costs may be arbitrarily correlated with the underlying data, how can we design mechanisms that are incentive-compatible, have robust learning guarantees, and optimize the cost-efficiency tradeoffs inherent in the learning problem?

This question is relevant to many real-world scenarios involving financial and strategic considerations in data procurement. Here are two examples:

(1) In the development of a certain drug, a pharmaceutical company wishes to train a disease classifier based on data obtained by hospitals and stored in patients' medical records. These data are not public, yet the company can offer hospital patients financial incentives to contribute their private records. We note the potential for cost heterogeneity: the compensation required by patients may be correlated with the content of their medical data (*e.g.* if they have the disease).

(2) Online retailers generally hope to know more about website visitors in order to better target products to customers. A retailer can offer to buy customers' demographic and social data, say in the form of access to their Facebook profile. But again, customers' willingness to sell may covary with their demographics data in an unknown way.

From sample complexity to budget efficiency

The classical problem in statistical learning theory is the following. We are given n datapoints (examples) $z_1, \dots, z_n \in \mathcal{Z}$ sampled from some distribution $\mathcal{D}$. Our goal is to select a hypothesis $h \in \mathcal{H}$ which "performs well" on unseen data from $\mathcal{D}$. We can specify performance in terms of a *loss function* $\ell(h, z)$, and we write $\mathcal{L}(h)$, known as the *risk* of h, to be the expectation of $\ell(h, z)$ on a random draw z from $\mathcal{D}$. The goal is to produce a hypothesis $\bar{h}$ whose risk is not much more than that of h^*, the optimal member of $\mathcal{H}$. For example, in *binary classification*, each data point consists of a pair $z = (x, y)$ where x encodes some "features" and $y \in \{-1, 1\}$ is the label; a hypothesis h is a function that predicts a label for a given set of features; and a typical loss function, the "0-1 loss", is defined so that $\ell(h, (x, y)) = 0$ when $h(x) = y$ and $\ell(h, (x, y)) = 1$ otherwise.

Research in statistical learning theory attempts to characterize how well such tasks can be performed in terms of the *resources* available and the inherent *difficulty* of the problem. The resource is usually the quantity of data n. In binary classification, for instance, the difficulty or richness of the problem is captured by the "VC-dimension" d, and a famous result [Vapnik 2000] is that there is an algorithm achieving the bound

$$\mathcal{L}(\bar{h}) \leq \mathcal{L}(h^*) + O\left(\sqrt{\frac{d \log n}{n}}\right), \tag{1}$$

with very high probability over the sample $z_1, \dots, z_n$.

In the present work we consider an alternative scenario: the learner has a fixed budget B and can use this budget to purchase examples. More precisely, on round t of a sequence of T rounds, agent t arrives with data point z_t, sampled i.i.d. from some $\mathcal{D}$, and a cost $c_t \in [0,1]$. This cost c_t is known only to the agent and can depend arbitrarily on z_t. The learning mechanism may offer a (possibly randomized) menu of *take-it-or-leave-it* prices $\pi_t : \mathcal{Z} \to \mathbb{R}_+$, with a possibly different price $\pi_t(z)$ for each data point z. The arriving agent observes the price $\pi_t(z_t)$ offered for her data and accepts as long as $\pi_t(z_t) \geq c_t$, in which case the mechanism pays the agent $\pi_t(z_t)$ and learns (c_t, z_t).[1] Our goal is to actively select prices to offer for different datapoints, subject to a budget B, in order to minimize the risk of our final output $\bar{h}$.

At a high level, our main result parallels the classical statistical learning guarantee in (1), but where the limited resource is the budget B instead of the sample size n.

MAIN RESULT 1 (INFORMAL). *For a large class of problems, there is an active data purchasing algorithm $\mathcal{A}$ that spends at most B in expectation and outputs a hypothesis $\bar{h}$ satisfying,*

$$\mathbb{E}\,\mathcal{L}(\bar{h}) \;\leq\; \mathcal{L}(h^*) + O\left(\sqrt{\frac{\gamma_{T,\mathcal{A}}}{B}}\right),$$

where $\gamma_{T,\mathcal{A}} \in [0,1]$ is an algorithm-dependent parameter of the (cost, data) sequence capturing the monetary difficulty of learning *and the expectation is over the algorithm's internal randomness.*

This bound depends on the quantity $\gamma_{T,\mathcal{A}}$ which captures the *monetary difficulty* of the problem at hand. (We also need as prior knowledge a rough estimate of $\gamma_{T,\mathcal{A}}$.) This is in rough analogy with VC-dimension in classical bounds such as Equation 1. Similarly, the key resource constraint is now the budget B rather than the quantity of data n.

It is important to note that $\gamma_{T,\mathcal{A}}$ depends on the choice of algorithm $\mathcal{A}$. However, our results also include simpler, algorithm-independent bounds. For instance, replace $\gamma_{T,\mathcal{A}}$ by $\sqrt{\mu}$, where μ is the mean of the arriving costs, and Main Result 1 continues to hold (and the only prior knowledge required is a rough estimate of μ). But $\gamma_{T,\mathcal{A}}$ can be significantly smaller than $\sqrt{\mu}$ when there are particular correlations between the costs and the examples; indeed, we can have $\gamma_{T,\mathcal{A}} \to 0$ even as μ stays constant. This indicates a case in which the average cost of data is high, but due to beneficial correlations between costs and data, our mechanism can obtain all the data it needs for good learning very cheaply. We give a thorough discussion of $\gamma_{T,\mathcal{A}}$ in Section 4.4.

Overview of Techniques

Our general idea for attacking this problem is to utilize online learning algorithms (OLAs) for regret minimization [Cesa-Bianchi and Lugosi 2006]. These algorithms output a hypothesis or prediction at each step $t = 1, \ldots, T$, and their performance is measured by the summed loss of these predictions over all the steps. The idea is that the hypotheses produced by the OLA at each step can be used both to determine the value of data during the procurement process and to generate a final prediction.

In Section 3, we lay out the tools we need for a pricing and learning mechanism to interact with OLAs. The first high-level problem is that, because of the budget constraint, our OLA will only see a small subset of the data sequence. We use the tool of *importance-weighting* to give good regret-minimization guarantees even when we do not see the entire data sequence. The second problem is how to aggregate the hypotheses of the OLA and convert its regret guarantee into a risk guarantee for our

[1] We will discuss the interaction model further in Sections 2 and 7.

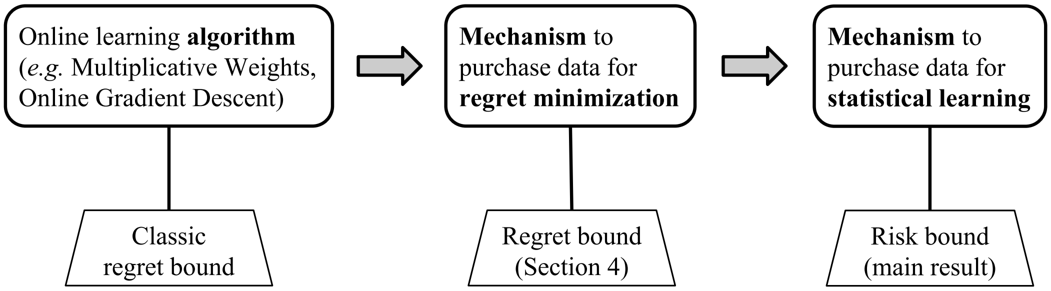

Online learning **algorithm**
(*e.g.* Multiplicative Weights, Online Gradient Descent)
Mechanism to purchase data for **regret minimization**
Mechanism to purchase data for **statistical learning**
Classic regret bound
Regret bound (Section 4)
Risk bound (main result)

Fig. 1: **Algorithmic and analytic approach.** First, we convert Follow-the-Regularized-Leader online no-regret algorithms into mechanisms that purchase data for a regret-minimization setting that we introduce for purposes of analysis. Then, we convert these into mechanisms to solve our main problem, statistical learning. The mechanisms interact with the online learning algorithms as black boxes, but the analysis relies on "opening the box".

statistical learning setting. This is achieved with the standard "online-to-batch" conversion [Cesa-Bianchi et al. 2004].

Given the tools of Section 3, the key remaining challenge is to develop a pricing and learning strategy that achieves low regret. We address this question in Section 4. We formally define a model of online learning for regret minimization with purchased data, in which the mechanism must output a hypothesis at each time step and perform well in hindsight against the entire data sequence, but only has enough budget to purchase and observe a fraction of the arriving data. We defer until later our detailed analysis of this setting, derivation of a pricing strategy, and lower bounds. At this point, we present our pricing strategy and regret guarantees for this setting.

In Section 5, we give our main results: risk guarantees for a learner with budget B and access to T arriving agents. These bounds follow directly by using the tools in Section 3 and regret-minimization results in Section 4.

In Section 6, we develop a deeper understanding of the regret minimization setting. We derive our pricing strategy from an in-depth analysis of a more analytically tractable variant of the problem, the "at-cost" setting, where the mechanism is only required to pay the cost of the arriving data point rather than the price posted. For this setting, we are able to derive the optimal pricing strategy for minimizing the regret bound of our class of learning algorithms subject to an expected budget constraint.

We also complement our upper bounds by proving lower bounds for data-purchasing regret minimization. These show that our mechanisms for the easier at-cost setting have an order-optimal regret guarantee of $\frac{T}{\sqrt{B}}\gamma_{T,\mathcal{A}}$. There is a small gap to our mechanisms for the main regret minimization setting, in which our guarantee is on the order of $\frac{T}{\sqrt{B}}\sqrt{\gamma_{T,\mathcal{A}}}$ (recall that $\gamma_{T,\mathcal{A}} \in [0,1]$, so this is a weaker guarantee). The dependence $T/\sqrt{B}$ approaches the classic $\sqrt{T}$ regret bound when B is large (approaching T). When B is small but still superconstant, we observe the perhaps counterintuitive fact that we can achieve $o(1)$ average regret per arrival while only observing an $o(1)$ fraction of the arriving data; in other words, we have "no data, no regret."

All proofs appear in the full version of the paper.

Related work

For "batch" settings in which all agents are offered a price simultaneously, pricing schemes for obtaining data have appeared in recent work, especially Roth and Schoenebeck [2012], which considered the design of mechanisms for efficient estimation of a statistic. However, this work and others in related settings [Cummings et al. 2015; Ghosh and Roth 2011; Ligett and Roth 2012] consider "passive" solutions that do not treat different types of data points differently, *e.g.* drawing a posted price independently regardless of the type of datapoint. We focus on an *active* approach in which the marginal value of individual examples is estimated according to the current learning progress and budget.

Horel et al. [2014] does take a data-dependent approach to purchasing data for learning, but focuses on a quite different learning setting, focusing on a model of regression with noisy samples and using a budget-feasible mechanism design approach.

Other works such as Dekel et al. [2008]; Ghosh et al. [2014]; Meir et al. [2012] focus on a setting in which agents may misreport their data (also see the peer-prediction literature). We suppose that agents may misreport their costs but not their data.

Many of the ideas in the present work draw from recent advances in using importance weighting for the active learning problem [Beygelzimer et al. 2009]. There is a wealth of theoretical research into active learning, including Balcan et al. [2006]; Beygelzimer et al. [2010]; Hanneke [2009] and many others.

"Budgeted Learning" is a somewhat related area of machine learning, but there the budget is not *monetary*. The idea is that we do not see all of the features of the data points in our set, but rather have a "budget" of the number of features we may observe (for instance, we may choose any two of the three features height, weight, age).

2. STATISTICAL LEARNING WITH PURCHASED DATA

In this section, we formally define the problem setting. The body of the paper will then consist of a series of steps for deriving mechanisms for this setting with provable guarantees, which will finally appear in Section 5.

We consider a statistical learning problem described as follows. Our data points are objects $z \in \mathcal{Z}$. We are given a hypothesis class $\mathcal{H}$ which we will assume is parameterized by vectors $\mathbb{R}^d$ but more broadly can be any Hilbert space endowed with a norm $\|\cdot\|$; for convenience we will treat elements $h \in \mathcal{H}$ as vectors which can be added, scaled, etc. We are also given a loss function $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$ that is convex in $\mathcal{H}$. We assume throughout the paper that the loss function is ***1-Lipschitz*** in h; that is, for any $z \in \mathcal{Z}$ and any $h, h' \in \mathcal{H}$ we have $|\ell(h, z) - \ell(h', z)| \leq \|h - h'\|$.

In many common scenarios, $\mathcal{Z}$ is the space of pairs (x, y) from the cross product $\mathcal{X} \times \mathcal{Y}$, with x the feature input and y the label, though in our setting $\mathcal{Z}$ can be a more generic object. For example, in the canonical problem of *linear regression*, we have that $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} = \mathbb{R}^d \times \mathbb{R}$, the hypothesis class is vectors $\mathcal{H} = \mathbb{R}^d$, and the loss function is defined according to squared error $\ell(h, (x, y)) := (h^\top x - y)^2$.

The **data-purchasing statistical learning problem** is parameterized by the data space $\mathcal{Z}$, hypothesis space $\mathcal{H}$, loss function ℓ, number of arriving data points T, and expected budget constraint B. A *problem instance* consists of a distribution $\mathcal{D}$ on the set $\mathcal{Z}$ and a sequence of pairs $(c_1, z_1), \ldots, (c_T, z_T)$ where each z_t is a data point drawn i.i.d. according to $\mathcal{D}$ and each $c_t \in [0, 1]$ is the private cost associated with that data point. The costs may be arbitrarily chosen, *i.e.* we consider a worst-case model of costs. (For instance, if costs and data are drawn together from a joint, correlated distribution, then this is a special case of our setting.)

In this problem, the task is to design a *mechanism* implementing the operations "post", "receive", and "predict" and interacting with the problem instance as follows.

— For each time step $t = 1, \ldots, T$:
  (1) The mechanism *posts* a pricing function $\pi_t : \mathcal{Z} \to \mathbb{R}$, where $\pi_t(z)$ is the price posted for data point z.
  (2) Agent t arrives, possessing (c_t, z_t).

(3) If the posted price $\pi_t(z_t) \geq c_t$, then agent t accepts the transaction: The mechanism pays $\pi_t(z_t)$ to the agent and *receives* (c_t, z_t). If $\pi_t(z_t) < c_t$, agent t rejects the transaction and the mechanism *receives* a null signal.

— The mechanism outputs a *prediction* $\bar{h} \in \mathcal{H}$.

Note that the mechanism is given the parameters $\mathcal{Z}$, $\mathcal{H}$, ℓ, T, and B, but the problem instance is completely unknown to the mechanism prior to to the arrivals. The design problem of the mechanism is how to choose the pricing function π_t to *post* at each time, how to update based on *receiving* data, and how to choose the final *prediction*. The *risk* or predictive error of a hypothesis is

$$\mathcal{L}(h) = \mathop{\mathbb{E}}_{z \sim \mathcal{D}} \ell(h, z)$$

and the goal of the mechanism is to minimize the risk $\mathcal{L}(\bar{h})$ of its final hypothesis $\bar{h}$. The benchmark is the optimal hypothesis in the class, $h^* = \arg\min_{h \in \mathcal{H}} \mathcal{L}(h)$.

The mechanism must guarantee that, for every input sequence $(c_1, z_1), \ldots, (c_T, z_T)$, it spends at most B in expectation over its own internal randomness.

Agent-mechanism interaction. The model of agent arrival and posted prices contains several assumptions. First, agents cannot fabricate data; they can only report data they actually have to the mechanism. Second, agents are rational in that they accept a posted price when it is higher than their cost and reject otherwise. Third, we have an implementation of the mechanism that can obtain the agent's cost c_t when the transaction occurs.

We emphasize that the purpose of this paper is not focused on the implementation of such a setting, but instead on developing active learning and pricing techniques and guarantees. This is also intended as a simple and clean model in which to begin developing such techniques. However, we briefly note some possible implementations.

In the most straightforward one, the mechanism posts prices directly to the agent who responds directly. This would be a weakly truthful implementation, as agents have no incentive to misreport costs after they choose to accept the transaction.

One strictly truthful implementation uses a *trusted third party* (TTP) that can facilitate the transactions (and guarantee the validity of the data if necessary). For example, we could imagine attempting to learn to classify a disease, and we could rely on a hospital to act as the broker allowing us to negotiate with patients for their data. Then the TTP/agent interaction could proceed as follows:

(1) Learning mechanism submits the pricing function π_t to the TTP;
(2) Agent provides his data point z_t and cost c_t to the TTP;
(3) TTP determines whether $\pi_t(z_t) \geq c_t$ and, if so, instructs the learner to pay $\pi_t(z_t)$ to the agent and then provides the pair (z_t, c_t) to the learner.

Other possibilities for strictly truthful implementation include using a bit of cryptography (see Section 7).

3. TOOLS FOR CONVERTING REGRET-MINIMIZING ALGORITHMS

In this section we begin with the classic regret-minimization problem and a broad class of algorithms for this problem. We then show how to apply techniques that convert these algorithms into a form that will be useful for solving the statistical learning problem with purchased data. The only missing ingredient will then be a price-posting strategy, which will be presented in Section 4.

3.1. Recap of classic regret-minimization setting and algorithms

In the classic regret-minimization problem, we have a hypothesis class $\mathcal{H}$ with the same assumptions as stated in Section 2. At each time $t = 1, \dots, T$ the algorithm posts a hypothesis $h_t \in \mathcal{H}$. Nature (the adversary, the environment, etc.) selects a 1-Lipschitz convex loss function $f_t : \mathcal{H} \to \mathbb{R}$.[2] The algorithm observes f_t and suffers loss $f_t(h_t)$.

The *loss* and *regret* of the algorithm on this particular input sequence are

$$\text{Regret}_T = \overbrace{\textstyle\sum_{t=1}^{T} f_t(h_t)}^{\text{Loss}_T} - \min_{h^* \in \mathcal{H}} \textstyle\sum_{t=1}^{T} f_t(h^*). \tag{2}$$

The regret objective is what one typically studies in adversarial settings, where we want to discount the loss incurred by the algorithm by the loss suffered by the best possible h^* chosen with knowledge of the sequence of f_t's. As we often consider randomized algorithms, we will generally consider *expected* loss and regret, where the expectation is over any randomness in the algorithm not over the (possibly-randomized) input sequence of loss functions. An algorithm is said to guarantee regret $R(T)$ if the latter provides an upper bound on regret for every sequence of loss functions $f_1, \dots, f_T$.

We utilize the broad class of *Follow-the-Regularized-Leader* (FTRL) online algorithms (Algorithm 1) [Shalev-Shwartz 2012; Zinkevich 2003]. Special cases of no-regret algorithms include Online Gradient Descent, Multiplicative Weights, and others. The algorithm requires a convex function $G : \mathcal{H} \to \mathbb{R}$ which is known as a *regularizer*.

ALGORITHM 1: Follow-the-Regularized-Leader (FTRL).

```
Input: learning parameter $\eta$, convex regularizer $G : \mathcal{H} \to \mathbb{R}$
for $t = 1, \dots, T$ do
    post hypothesis $h_t$, observe loss function $f_t$;
    update $h_{t+1} = \inf_{h \in \mathcal{H}} \left\{ \sum_{t' \leq t} f_{t'}(h) + \eta G(h) \right\}$;
end
```

It is well-known (and indeed follows as a special case of Lemma 3.2) that, under the assumptions on our setting, FTRL algorithms guarantee an expected regret bound of $O(\sqrt{T})$, and this is tight with respect to T.

3.2. Importance-Weighting Technique for Less Data

As a starting point, suppose we wish to design an online learning algorithm that does not observe all of the arriving loss functions, but still performs well against the entire arrival sequence.

Because the arrival sequence may be adversarially chosen, a good algorithm should randomly choose to sample some of the arrivals. In this section, we abstract away the decision of how to randomly sample. (This will be the focus of Section 4.) In this section, we suppose that at each time t, after posting a hypothesis h_t, a probability $q_t > 0$ is specified by some external means as a (possibly random) function of the preceding time steps. With probability q_t, we observe f_t; with probability $1 - q_t$, we observe nil.

Our goal is to modify our online learning algorithms for this setting and obtain a modified regret guarantee. Notice crucially that the definition of loss and regret (2) are unchanged: We still suffer the loss $f_t(h_t)$ regardless of whether we observe f_t.

[2] This definition of "loss function" is a departure from our main setting which involved $\ell(\cdot, \cdot)$. But we will use this somewhat more general setup by choosing $f_t(h) \propto \ell(h, z_t)$ for the datapoint z_t.

The key technique we use is *importance weighting*. The idea is that, if we only observe each of a sequence of values x_i with probability p_i, then we can get an unbiased estimate of their sum by taking the sum of $\frac{x_i}{p_i}$ for those we do observe. To check this fact, let $\mathbb{1}_i$ be the indicator variable for the event that we observe i and note that the expectation of our sum is $\mathbb{E}\left[\sum_i \mathbb{1}_i \frac{x_i}{p_i}\right] = \sum_i x_i$. This is called importance-weighting the observations (and is a specific instance of a more general machine learning technique). Furthermore, if each $\frac{x_i}{p_i}$ is bounded and observed independently, we can expect the estimate to be quite good via tail bounds.

The importance-weighted modification to an online learning algorithm is outlined in Algorithm 2. The importance-weighted regret guarantee we obtain is given in Lemma 3.2. It depends on the following key notation. Our analysis and algorithm require a given norm $\|\cdot\|$, and we recall the definition of the dual norm $\|z\|_\star := \sup_{x;\|x\|\le 1} x \cdot z$.

Definition 3.1. Given $h \in H$, and convex loss $f : \mathcal{H} \to \mathbb{R}$, let $\Delta_{h,f} := \|\nabla f(h)\|_\star$.

We can informally think of $\Delta_{h,f}$ both as the "difficulty" of arrival f when the current hypothesis is h, and as the "value" of observing f. This interpretation is explored in Section 4 when we define the parameter $\gamma_{T,\mathcal{A}}$.

ALGORITHM 2: Importance-Weighted Online Learning Algorithm.

```
Input: access to Online Learning Algorithm (OLA)
for $t = 1, \dots, T$ do
    post hypothesis $h_t \leftarrow$ OLA; observe sampling probability $q_t$;
    toss $q_t$-weighted coin (Bernoulli sample) $\epsilon_t$ ;
    if $\epsilon_t = \begin{cases} 1 & \text{input importance-weighted loss function} \quad \hat{f}_t(\cdot) = \frac{f_t(\cdot)}{q_t} \to \text{OLA} \\ 0 & \text{input zero function} \quad \hat{f}_t(\cdot) \equiv 0 \to \text{OLA} \end{cases}$
end
```

LEMMA 3.2. *Assume we implement Algorithm 2 with nonzero sampling probabilities $q_1, \dots, q_T$. Assume the underlying OLA is FTRL (Algorithm 1) with regularizer $G : \mathcal{H} \to \mathbb{R}$ that is strongly convex with respect to $\|\cdot\|$. Then the expected regret, with respect to the loss sequence $f_1, \dots, f_T$, is no more than*

$$R(T) = \frac{\beta}{\eta} + 2\eta\, \mathbb{E}\left[\sum_{t=1}^{T} \frac{\Delta^2_{h_t,f_t}}{q_t}\right],$$

where β is a constant depending on $\mathcal{H}$ and G, η is a parameter of the algorithm, and the expectation is over any randomness in the choices of h_t and q_t.

We can recover the classic regret bound as follows: Take each $q_t = 1$, and note by the Lipschitz assumption that each $\Delta_{h_t,f_t} \le 1$. Then by setting $\eta = \Theta(1/\sqrt{T})$, we get an expected regret bounded by $O(\sqrt{T})$.

3.3. The "Online-to-Batch" Conversion

So far so good: We can convert an online regret-minimization algorithm to use smaller amounts of data, and we postpone the question of how to price data till Section 4. We now address the statistical learning problem, which is how to generate accurate *predictions* based on the online learning process.

We address this with a standard tool known as the "online-to-batch conversion," where we may leverage an online learning algorithm for use in a "batch" setting. A sketch of this technique is as follows, and further details can be found in, e.g., Shalev-Shwartz [2012]. Given a batch of i.i.d. data points, feed them one-by-one into the no-regret algorithm. Because the algorithm has low regret, its hypotheses predicted well on average. But since each data point was drawn i.i.d., this means that these hypotheses on average predict well on an i.i.d. draw from the distribution. Thus it suffices to take the mean of the hypotheses to obtain low risk.

LEMMA 3.3 (ONLINE-TO-BATCH [CESA-BIANCHI ET AL. 2004]). *Suppose the sequence of convex loss functions* $f_1, \ldots, f_T$ *are drawn i.i.d. from a distribution* $\mathcal{F}$ *and that an online learning algorithm with hypotheses* $h_1, \ldots, h_T$ *achieves expected regret* $R(T)$*. Let* $\mathcal{L}(h) = \mathbb{E}_{f \sim \mathcal{F}} f(h)$ *and* $h^* = \arg\min_{h \in \mathcal{H}} \mathcal{L}(h)$*. For* $\bar{h}_{1:T} = \frac{1}{T}\sum_{t=1}^{T} h_t$*, we have*

$$\mathop{\mathbb{E}}_{\substack{f_1,\ldots,f_T,\\ alg}} \mathcal{L}(\bar{h}_{1:T}) \leq \mathcal{L}(h^*) \; + \; \frac{1}{T}R(T).$$

We note that this conversion will continue to hold in the data-purchasing no-regret setting we define next, since all that is required is that the algorithm output a hypothesis h_t at each step and that there is a regret bound on these hypotheses.

4. REGRET MINIMIZATION WITH PURCHASED DATA

In this setting, we define the problem of regret minimization with purchased data. We will design mechanisms with good regret guarantees for this problem, which will translate via the aforementioned online-to-batch conversion (Lemma 3.3) into guarantees for our original problem of statistical prediction.

The essence of the data-purchasing no-regret learning setting is that an online algorithm ("mechanism") is asked to perform well against a sequence of data, but by default, the mechanism does not have the ability to see the data. Rather, the mechanism may purchase the right to observe data points using a limited budget. The mechanism is still expected to have low regret compared to the optimal hypothesis in hindsight on the entire data sequence (even though it only observes a portion of the sequence).

4.1. Problem Definition

The **data-purchasing regret minimization problem** is parameterized by the hypothesis space $\mathcal{H}$, number of arriving data points T, and expected budget constraint B. A *problem instance* is a sequence of pairs $(c_1, f_1), \ldots, (c_T, f_T)$ where each $f_t : \mathcal{H} \to \mathbb{R}$ is a convex loss function and each $c_t \in [0, 1]$ is the cost associated with that data point. We assume that the f_t are 1-Lipschitz, and let $\mathcal{F}$ be the set of such loss functions.

In this problem, we design a *mechanism* implementing the operations "post" and "receive" and interacting with the problem instance as follows.

— For each time step $t = 1, \ldots, T$:
  (1) The mechanism *posts* a hypothesis h_t and a pricing function $\pi_t : \mathcal{F} \to \mathbb{R}$, where $\pi_t(f)$ is the price posted for loss function f.
  (2) Agent t arrives, possessing (c_t, f_t).
  (3) If the posted price $\pi_t(f_t) \geq c_t$, then agent t accepts the transaction: The mechanism pays $\pi_t(f_t)$ to the agent and *receives* (c_t, f_t). If $\pi_t(f_t) < c_t$, agent t rejects the transaction and the mechanism *receives* a null signal.

Note the key differences from the statistical learning setting: We must post a hypothesis h_t at each time step (and we do not output a final prediction), and data is not assumed to come from a distribution.

The goal of the mechanism is to minimize the loss, namely $\sum_t f_t(h_t)$. The definition of regret is also the same as in the classical setting (Equation 2). Note that we suffer a loss $f_t(h_t)$ at time t regardless of whether we purchase f_t or not. The mechanism must also guarantee that, for every problem instance $(c_1, f_1), \ldots, (c_T, f_T)$, it spends at most B in expectation over its own internal randomness.

4.2. The Importance-Weighting Framework

Recall that, in Section 3.2, we introduced the *importance-weighting* technique for online learning. This gave regret guarantees for a learning algorithm when each arrival f_t is observed with some probability q_t.

Our general approach will be to develop a strategy for randomly drawing posted prices π_t. This will induce a probability q_t of obtaining each arrival f_t.

Therefore, the entire problem has been reduced to choosing a posted-price strategy at each time step. This posted-price strategy should attempt to minimize the regret bound while satisfying the expected budget constraint.

A brief sketch of the proof arguments is as follows. After we choose a posted price strategy, each q_t will be determined as a function of h_t, c_t, and f_t. (q_t is just equal to the probability that our randomly drawn price exceeds the agent's cost c_t.) Thus, we can apply Lemma 3.2, which stated that for these induced probabilities q_t, the expected regret of the learning algorithm is

$$\frac{\beta}{\eta} + 2\eta \, \mathbb{E} \sum_t \frac{\Delta^2_{h_t,f_t}}{q_t},$$

where β is a constant and η is a parameter of the learning algorithm to be chosen later.

After we choose and apply such a strategy, the general approach to proving our regret bounds is to find an *a priori* bound M such that $2\,\mathbb{E}\sum_t \frac{\Delta^2_{h_t,f_t}}{q_t} \leq M$. Then the regret bound becomes $\frac{\beta}{\eta} + \eta M$. If we know this upper-bound M in advance using some prior knowledge, then we can choose $\eta = \Theta(1/\sqrt{M})$ as the parameter for our learning algorithms. This gives a regret guarantee of $O(\sqrt{M})$.

4.3. A First Step to Pricing: The "At-Cost" Variant

The bulk of our analysis of the no-regret data-purchasing problem actually focuses on a slightly easier variant of the setting: If the arriving agent accepts the transaction, then the mechanism only has to pay the cost c_t rather than the posted price $\pi_t(f_t)$. We call this the "at-cost" variant of the problem. This setting turns out to be much more analytically tractable: We derive optimal regret bounds for our mechanisms and matching lower bounds. We then take the key approach and insights derived from this variant and apply them to produce a solution to the main no-regret data purchasing problem. In order to keep the story moving forward, we summarize our results for the "at-cost" setting here and explore how they are obtained in Section 6.

In the at-cost setting, we are able to solve directly for the pricing strategy that minimizes the importance-weighted regret bound of Lemma 3.2. We first define one important quantity, then we state the strategy and result in Theorem 4.2.

Definition 4.1. For a fixed input sequence $(c_1, f_1), \ldots, (c_T, f_T)$, $\Delta_{h,f}$ in Definition 3.1, and a mechanism outputting (possibly random) hypotheses $h_1, \ldots, h_T$, define

$$\gamma_{T,\mathcal{A}} = \mathbb{E}\,\frac{1}{T}\sum_t \Delta_{h_t,f_t}\sqrt{c_t}$$

where the expectation is over the randomness of the algorithm. Note that $\gamma_{T,\mathcal{A}}$ lies in $[0,1]$ by our assumptions on bounded cost and Lipschitz loss.

Now we give the main result for the at-cost setting.

THEOREM 4.2. *There is a mechanism for the "at-cost" problem of data purchasing for regret minimization that interfaces with FTRL and guarantees to meet the expected budget constraint, where for a parameter $\gamma_{T,\mathcal{A}} \in [0,1]$ (Definition 4.1),*

(*1*) *The expected regret is bounded by $O\left(\max\left\{\frac{T}{\sqrt{B}}\gamma_{T,\mathcal{A}}\ ,\ \sqrt{T}\right\}\right)$.*
(*2*) *This is optimal in that no mechanism can improve beyond constant factors.*
(*3*) *The pricing strategy is to choose a parameter $K = O\left(\frac{T}{B}\gamma_{T,\mathcal{A}}\right)$ and draw $\pi_t(f)$ randomly according to a distribution such that $\Pr[\pi_t(f) \geq c] = \min\left\{1\ ,\ \frac{\Delta_{h_t,f}}{K\sqrt{c}}\right\}$.*

The only prior knowledge required is an estimate of $\gamma_{T,\mathcal{A}}$ up to a constant factor.

4.4. Interpreting the Quantity $\gamma_{T,\mathcal{A}}$

Several of our bounds rely heavily on the quantity $\gamma_{T,\mathcal{A}}$ which measures, in a sense, the "financial difficulty" of the problem. We now devote some discussion to understanding $\gamma_{T,\mathcal{A}}$ by answering four questions.

(1) How to interpret $\gamma_{T,\mathcal{A}}$?

$\gamma_{T,\mathcal{A}}$ is an average, over time steps t, of $\Delta_{h_t,f_t} \cdot \sqrt{c_t}$. Here, Δ_{h_t,f_t} intuitively captures both the "difficulty" of the data f_t and also the "value" or "benefit" of f_t. To explain the difficulty aspect, by examining the regret bound for FTRL learning algorithms (*e.g.* the importance-weighted regret bound of Lemma 3.2 with all $q_t = 1$), one observes that if each Δ_{h_t,f_t} is small, then we have an excellent regret bound for our learning algorithm; the problem is "easy". To explain the value aspect, one can for concreteness take the Online Gradient Descent algorithm; the larger the gradient, the larger the update at this step, and Δ_{h_t,f_t} is the norm of the gradient. And in general, the higher Δ_{h_t,f_t}, the more likely we are to purchase arrival f_t.

Thus, $\gamma_{T,\mathcal{A}}$ captures the correlations between the value of the arriving data and the cost of that data. If either the mean of the costs or the average benefit Δ_{h_t,f_t} of the data is converging to 0, then $\gamma_{T,\mathcal{A}} \to 0$ and in these cases we can learn with high accuracy very cheaply, as may be expected. More interestingly, it is possible to have both high average costs, and high average data-values, and yet still have $\gamma_{T,\mathcal{A}} \to 0$ due to beneficial correlations. In these cases we can learn much more cheaply than might be expected based on either the economic side or the learning side alone.

(2) When should we expect to have good prior knowledge of $\gamma_{T,\mathcal{A}}$?

Although in general $\gamma_{T,\mathcal{A}}$ will be domain-specific, there are several reasons for optimism. First, $\gamma_{T,\mathcal{A}}$ compresses all information about the data and costs into a single scalar parameter (compare to the common mechanism-design assumption that the prior distribution of agents' values is fully known). Second, we do not need very exact estimates of $\gamma_{T,\mathcal{A}}$ (*e.g.* we do not need to know $\gamma_{T,\mathcal{A}} \pm \epsilon$): For order-optimal regret bounds, we only need an estimate within a constant factor of $\gamma_{T,\mathcal{A}}$. Third, $\gamma_{T,\mathcal{A}}$ is directly proportional to K, which is a normalization constant in our pricing distribution: If we increase K, the probability of obtaining a given data point only decreases, and vice versa. In fact, the best choice of K is the normalization constant so that we run out of budget precisely when the last arrival leaves. Thus, K (equivalently, $\gamma_{T,\mathcal{A}}$) can be estimated and adjusted online by tracking the "burn rate" (spending per unit time) of the algorithm. In simulations (appearing in the full version of the paper), we have observed success with a simple approach of estimating K based on the average correlation so far along with the burn rate, *i.e.* if the current estimated $\gamma_{T,\mathcal{A}}$ is $\hat{\gamma_{T,\mathcal{A}}}$ and there are $\hat{T}$ steps remaining with $\hat{B}$ budget remaining to spend, set $K = \hat{\gamma_{T,\mathcal{A}}}\hat{T}/\hat{B}$.

(3) What can we prove without prior knowledge of $\gamma_{T,\mathcal{A}}$?

It turns out that, if we only have an estimate of $\bar{c} = \frac{1}{T}\sum_t \sqrt{c_t}$, respectively $\mu = \frac{1}{T}\sum_t c_t$, then this suffices for regret guarantees on the order of $T\bar{c}/\sqrt{B}$, respectively $T\sqrt{\mu}/\sqrt{B}$. This "graceful degradation" will continue to be true in the main setting. The idea is that we can follow the optimal form of the pricing strategy while choosing any normalization constant $K \geq \frac{T}{B}\gamma_{T,\mathcal{A}}$. It may no longer be optimal, but it will ensure that we satisfy the budget and give guarantees depending on the magnitude of K. So all we need is an approximate estimate of some value larger than $\gamma_{T,\mathcal{A}}$. Both $\bar{c}$ and μ are guaranteed to upper-bound on $\gamma_{T,\mathcal{A}}$, so both can be used to pick K while satisfying the budget.

To recap, knowledge of only a simple statistic such as the mean of the arriving costs suffices for good learning guarantees, with better knowledge translating to better guarantees.

(4) $\gamma_{T,\mathcal{A}}$ depends on the algorithm—what are the implications?

We first note that $\gamma_{T,\mathcal{A}}$ can be upper-bounded by, for instance, $\sqrt{\mu}$ where μ is the average of the arriving costs. So a bound containing $\gamma_{T,\mathcal{A}}$ does imply nontrivial algorithm-independent bounds. The purpose of $\gamma_{T,\mathcal{A}}$ is to capture cases where we can do significantly better than such bounds. It captures improvement due to good correlations between cost and data and/or due to the learning algorithm being a good fit for the problem. If one chooses a good algorithm for the problem, the dependence on $\gamma_{T,\mathcal{A}}$ shows that the mechanism can take advantage of correlations when they exist (for instance, when the lower-cost points are also the most beneficial for learning).

However, one might hope for an algorithm-independent quantity that, in analogy with VC-dimension, captures the "difficulty" of the purchasing and learning problem instance. This leads to the question:

(4a) Can we remove the algorithm-dependence of the bound? One might hope to achieve a bound depending on an algorithm-independent quantity that captures correlations between data and cost. A natural candidate is $\gamma^*_{T,\mathcal{A}} := \frac{1}{T}\sum_t \Delta_{h^*,f_t}\sqrt{c_t}$. In general, there are difficult cases where one can not achieve a bound in terms of $\gamma^*_{T,\mathcal{A}}$. However, in nicer scenarios we may expect $\gamma_{T,\mathcal{A}}$ to approximate $\gamma^*_{T,\mathcal{A}}$. For instance, suppose $\ell(h,z) = \phi(h^\top z)$ where ϕ is a differentiable convex function whose gradient is 1-Lipschitz — commonly-used examples include the *squared hinge loss* and the *log loss*. Under this condition, where again we are using $f_t(\cdot) := \ell(\cdot, z_t)$, we can show that

$$\begin{aligned}
\Delta_{h_t,f_t}\sqrt{c_t} - \Delta_{h^*,f_t}\sqrt{c_t} &= \|\nabla\ell(h_t,z_t)\|_\star\sqrt{c_t} - \|\nabla\ell(h^*,z_t)\|_\star\sqrt{c_t} \\
&\leq \|(\phi'(h_t^\top z_t) - \phi'(h^{*\top} z_t))z_t\|_\star \\
&\leq |\phi(h_t^\top z_t) - \phi(h^{*\top} z_t)| = |\ell(h_t,z_t) - \ell(h^*,z_t)|.
\end{aligned}$$

By the regret guarantee of our mechanism when run with a good algorithm, even initialized with very weak knowledge, this difference in losses per time step is $o(1)$, implying that $\gamma_{T,\mathcal{A}} \to \gamma^*_{T,\mathcal{A}}$. A deeper investigation of this phenomenon is a good candidate for future work.

4.5. Mechanisms and Results for Data-Purchasing Regret Minimization

Unlike in the "at-cost" setting, we cannot in general solve for the form of the optimal pricing strategy. This is intuitively because, when we must pay the price we post, the optimal strategy depends on c_t. But the algorithm cannot condition the purchasing decision directly on c_t, as this is private information of the arriving agent.

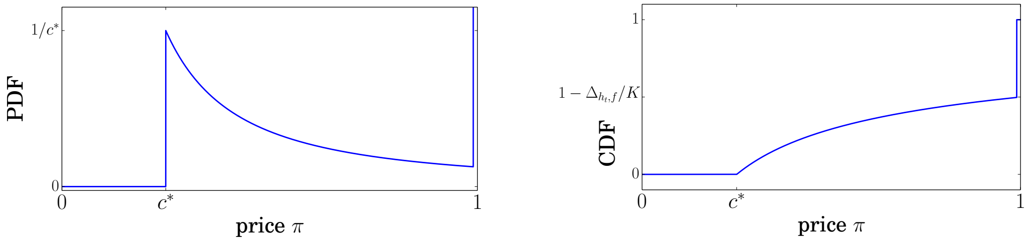
We propose simply drawing posted prices according to the optimal strategy derived for the at-cost setting, namely,

$$\mathbf{Pr}[\pi_t(f) \geq c] = \min\left\{1\ ,\ \frac{\Delta_{h_t,f}}{K\sqrt{c}}\right\}, \tag{3}$$

but with a different choice of normalization constant K. We note that there is a pricing distribution that accomplishes this:

OBSERVATION 1. *For any K and $\Delta_{h_t,f}$, there exists a pricing distribution on $\pi_t(f)$ that satisfies Equation 3. Letting $c^* = \Delta^2_{h_t,f}/K^2$, the CDF is given by $F(\pi) = \mathbf{Pr}[\pi_t(f) \leq \pi] = 0$ if $\pi \leq c^*$, $F(\pi) = 1 - \Delta_{h_t,f}/K\sqrt{\pi}$ if $c^* \leq \pi \leq 1$, and $F(\pi) = 1$ if $\pi > 1$.*

The pricing distribution is given in Figure 2. This strategy gives Mechanism 3.



(a) Probability density function of the pricing distribution. The price $\pi(f) = 1$ with probability $\min\{1, \Delta_{h_t,f}/K\}$. On the interval $(c^*, 1)$ the density function is $x \mapsto \Delta_{h_t,f}/2Kx^{3/2}$.

(b) Cumulative distribution function of the pricing distribution. Equal to zero for $\pi \leq c^*$, then equal to $1 - \Delta_{h_t,f}/K\sqrt{\pi}$ on $(c^*, 1)$, then equal to 1 at cost 1.

Fig. 2: **The pricing distribution.** Illustrates the distribution from which we draw our posted prices at time t, for a fixed arrival f. The quantity $\Delta_{h_t,f}$ captures the "benefit" from obtaining f. K is a normalization parameter. The distribution's support has a lowest price c^*, which has the form $c^* = \Delta^2_{h_t,f_t}/K^2$.

Mechanism 3: Mechanism for no-regret data-purchasing problem.

```
Input: parameters $K$, $\eta$, access to online learning algorithm (OLA)
set OLA parameter $\eta$;
for $t = 1, \ldots, T$ do
    post hypothesis $h_t \leftarrow$ OLA;
    post prices $\pi_t(f)$ drawn randomly such that $\mathbf{Pr}[\pi_t(f) \geq c] = \min\left\{1\ ,\ \frac{\Delta_{h_t,f}}{K\sqrt{c}}\right\}$;
    if we receive $(c_t, f_t)$ then
        let $q_t = \mathbf{Pr}_{\pi_t}[\pi_t(f_t) \geq c_t]$;
        let importance-weighted loss function $\hat{f}_t(\cdot) = \frac{f_t(\cdot)}{q_t}$;
        send $\hat{f}_t \rightarrow$ OLA;
    else
        send $0$ function $\rightarrow$ OLA;
    end
end
```

As in the known-costs case, our regret bounds depend upon the prior knowledge of the algorithm. It will turn out to be helpful to have prior knowledge about both $\gamma_{T,\mathcal{A}}$ and the following parameter, which can be interpreted as $\gamma_{T,\mathcal{A}}$ with all costs $c_t = 1$:

$$\gamma_{T,\mathcal{A}}^{\max} = \mathbb{E}\,\frac{1}{T}\sum_t \Delta_{h_t,f_t}.$$

THEOREM 4.3. *If Mechanism 3 is run with prior knowledge of $\gamma_{T,\mathcal{A}}$ and of $\gamma_{T,\mathcal{A}}^{\max}$ (up to a constant factor), then it can choose K and η to satisfy the expected budget constraint and obtain a regret bound of*

$$O\left(\max\left\{\frac{T}{\sqrt{B}}g\ ,\ \sqrt{T}\right\}\right),$$

where $g = \sqrt{\gamma_{T,\mathcal{A}} \cdot \gamma_{T,\mathcal{A}}^{\max}}$ (by setting $K = \frac{T}{B}\gamma_{T,\mathcal{A}}^{\max}$). Similarly, knowledge only of $\gamma_{T,\mathcal{A}}$, respectively $\bar{c} = \frac{1}{T}\sum_t \sqrt{c_t}$, respectively $\mu = \frac{1}{T}\sum_t c_t$ suffices for the regret bound with $g = \sqrt{\gamma_{T,\mathcal{A}}}$, respectively $g = \sqrt{\bar{c}}$, respectively $g = \mu^{1/4}$.

We can observe a quantifiable "price of strategic behavior" in the difference between the regret guarantees of Theorems 4.3 (this setting) and Theorem 4.2 (the "at-cost") setting:

$$\frac{T}{\sqrt{B}}\sqrt{\gamma_{T,\mathcal{A}} \cdot \gamma_{T,\mathcal{A}}^{\max}} \quad \text{vs} \quad \frac{T}{\sqrt{B}}\gamma_{T,\mathcal{A}}.$$

Note that $\gamma_{T,\mathcal{A}}^{\max} \geq \gamma_{T,\mathcal{A}}$, and they approach equality as all costs approach the upper bound 1, but become very different as the average cost $\mu \to 0$ while the maximum cost remains fixed at 1.

Comparison to lower bound. Our lower-bound for the data purchasing regret minimization problem is $\Omega\left(\frac{T}{\sqrt{B}}\gamma_{T,\mathcal{A}}\right)$ (because the lower bound for the at-cost setting, mentioned in Theorem 4.2 and stated in Theorem 6.3, applies to the general setting as well). So the difference in bounds discussed above, a factor of $\sqrt{\gamma_{T,\mathcal{A}}}$ versus $\sqrt{\gamma_{T,\mathcal{A}}^{\max}}$, is the only gap between our upper and lower bounds for the general data purchasing no regret problem.

The most immediate open problem in this paper is close this gap. Intuitively, the lower bound does not take advantage of "strategic behavior" in that a posted-price mechanism may often have to pay significantly more than the data actually costs, meaning that it obtains less data in the long run. Meanwhile, it may be possible to improve on our upper-bound strategy by drawing prices from a different distribution.

5. RESULTS FOR STATISTICAL LEARNING

In this section, we give the final mechanism, Mechanism 4, for the data purchasing statistical learning problem. The idea is to simply run the regret-minimization Mechanism 3 on the arriving agents. At each stage, Mechanism 3 posts a hypothesis h_t. We then aggregate these hypothesis by averaging to obtain our final prediction.

THEOREM 5.1. *Mechanism 4 guarantees spending at most B in expectation and*

$$\mathbb{E}\,\mathcal{L}(\bar{h}) \leq \mathcal{L}(h^*) + O\left(\max\left\{\tfrac{g}{\sqrt{B}}\ ,\ \sqrt{\tfrac{1}{T}}\right\}\right),$$

where $g = \sqrt{\gamma_{T,\mathcal{A}} \cdot \gamma_{T,\mathcal{A}}^{\max}}$, assuming that $\gamma_{T,\mathcal{A}}$ and $\gamma_{T,\mathcal{A}}^{\max}$ are known in advance up to a constant factor.

Mechanism 4: Mechanism for statistical learning data-purchasing problem.

```
Input: parameters $K, \eta$, access to OLA
identify each data point $z$ with the loss function $f(\cdot) = \ell(\cdot, z)$;
run Mechanism 3 with parameters $\eta, K$ and access to OLA;
let $h_1, \ldots, h_T$ be the resulting hypotheses;
output $\bar{h} = \frac{1}{T}\sum_t h_t$;
```

If one assumes approximate knowledge respectively of $\gamma_{T,\mathcal{A}}$, of $\bar{c} = \frac{1}{T}\sum_t \sqrt{c_t}$, or of $\mu = \frac{1}{T}\sum_t c_t$, then the guarantee holds with respectively $g = \sqrt{\gamma_{T,\mathcal{A}}}$, $g = \sqrt{\bar{c}}$, or $g = \mu^{1/4}$.

PROOF. By Theorem 4.3, Mechanism 3 guarantees an expected regret of $O\left(\max\left\{\frac{T}{\sqrt{B}}g\ ,\ \sqrt{T}\right\}\right)$ when run with the specified prior knowledge for the specified values of g. Therefore, the online-to-batch conversion of Lemma 3.3 proves the theorem. □

The statement of Main Result 1 is the special case where only $\gamma_{T,\mathcal{A}}$ is known and $g = \sqrt{\gamma_{T,\mathcal{A}}}$. A detailed discussion of $\gamma_{T,\mathcal{A}}$ is in Section 4.4.

6. DERIVING PRICING AND THE “AT-COST” VARIANT

In this section, we dig a bit deeper into the no-regret data purchasing problem, deriving our posted-price distribution and also our lower bounds. To do so, we consider a simpler “at-cost” variant of the no-regret data purchasing problem defined in Section 4: Rather than paying the price it posts, the mechanism is only required to pay the cost c_t of the data it is acquiring. Otherwise, the setting is exactly the same.

We first show how our posted-price strategy is derived as the optimal solution to the at-cost problem of minimizing regret subject to the budget constraint. The resulting upper bounds for the “at-cost” variant were given in Theorem 4.2. Then, we give some fundamental lower-bounds on regret, showing that in general our upper bounds cannot be improved upon here. These lower bounds also hold for the main no-regret data purchasing problem.

6.1. Deriving an Optimal Pricing Strategy

We begin by asking what seems to be an even easier question. Suppose that for every pair (c_t, f_t) that arrives, we could first “see” (c_t, f_t), then choose a probability with which to obtain (c_t, f_t) and pay c_t. What would be the optimal probability with which to take this data?

LEMMA 6.1. *To minimize the regret bound of Lemma 3.2, the optimal choice of sampling probability is of the form $q_t = \min\{1\ ,\ \Delta_{h_t,f_t}/K^*\sqrt{c_t}\}$. The normalization factor $K^* \approx \frac{T}{B}\gamma_{T,\mathcal{A}}$.*

The proof follows by formulating the convex programming problem of minimizing the regret bound of Lemma 3.2 subject to an expected budget constraint. It also gives the form of the normalization constant K^*, which depends on the input data sequence and the hypothesis sequence.

The key insight is now that we can actually achieve the sampling probabilities dictated by Lemma 6.1 using a randomized posted-price mechanism. That is, by randomly drawing posted prices according to our distribution, we choose to purchase (c_t, f_t) with exactly the probability q_t stated in Lemma 6.1, for any possible value of c_t and without knowing (c_t, f_t). To see this, once we fix f_t, notice that the optimal sampling probability is decreasing in the associated c_t. Thus, for each possible f (including the f_t that actually arrives), by drawing a posted price from the appropriate distribution, we induce for all possible c_t the required probability of purchase.

Thus, our final mechanism for the at-cost variant is to simply apply Mechanism 3, but only pay the cost of the arrival rather than the price we posted. We set $K = \frac{T}{B}\gamma_{T,\mathcal{A}}$. Note that this choice of normalization constant K is different from the main setting because we on average pay less in the at-cost setting; this leads to the difference in the regret bounds. Our main bound for the at-cost variant was given in Theorem 4.2. An open problem for this setting is whether one can obtain the same regret bounds without any prior knowledge at all about the arriving costs and data.

6.2. Lower bounds for data purchasing regret minimization

Here, we prove lower bounds analogous to the classic regret lower bound, which states that no algorithm can guarantee to do better than $O(\sqrt{T})$. These lower bounds will hold even in the "at-cost" setting, where they match our upper bounds. An open problem is to obtain a larger-order lower bound for the main setting where the mechanism pays its posted price. This would show a separation between the at-cost variant and the main problem.

First, we give what might be considered a "sample complexity" lower bound for no-regret learning: It specializes our setting to the case where all costs are equal to one (and this is known to the algorithm in advance), so the question is what regret is achievable by an algorithm that observes B of the T arrivals.

THEOREM 6.2. *Suppose all costs $c_t = 1$. No algorithm for the at-cost online data-purchasing problem has regret better than $O(T/\sqrt{B})$; that is, for every algorithm, there exists an input sequence on which its regret is $\Omega(T/\sqrt{B})$.*

Proof Idea: We will have two coins, with probabilities $\frac{1}{2} \pm \epsilon$ of coming up heads. We will take one of the coins and provide T i.i.d. flips as the input sequence. The possible hypotheses for the algorithm are $\{\text{heads}, \text{tails}\}$ and the loss is zero if the hypothesis matches the flip and one otherwise. The cost of every data point will be one.

The idea is that an algorithm with regret much smaller than $T\epsilon$ must usually predict heads if it is the heads-biased coin and usually predict tails if it is the tails-biased coin. Thus, it can be used to distinguish these cases. However, there is a lower bound of $\Omega\left(\frac{1}{\epsilon^2}\right)$ samples required to distinguish the coins, and the algorithm only has enough budget to gain information about $O(B)$ of the samples. Setting $\epsilon = 1/\sqrt{B}$ gives the regret bound. □

We next extend this idea to the case with heterogeneous costs. The idea is very simple: Begin with the problem from the label-complexity lower bound, and introduce "useless" data points and heterogeneous costs. The worst or "hardest" case for a given average cost is when cost is perfectly correlated with benefit, so all and only the "useful" data points are expensive.

THEOREM 6.3. *No algorithm for the non-strategic online data-purchasing problem has expected regret better than $O\left(\gamma_{T,\mathcal{A}}T/\sqrt{B}\right)$; that is, for every $\gamma_{T,\mathcal{A}}$, for every algorithm, there exists a sequence with parameter $\gamma_{T,\mathcal{A}}$ on which its regret is $\Omega\left(\gamma_{T,\mathcal{A}}T/\sqrt{B}\right)$. Similarly, for $\bar{c} = \frac{1}{T}\sum_t \sqrt{c}$ and $\mu = \frac{1}{T}\sum_t c_t$, we have the lower bounds $\Omega\left(T\bar{c}/\sqrt{B}\right)$ and $\Omega\left(T\sqrt{\mu}/\sqrt{B}\right)$.*

7. DISCUSSION AND CONCLUSION

7.1. Agent-Mechanism Interaction Model

Our model of interaction, while perhaps the simplest initial starting point, involves some subtleties that may be interesting to address in the future. A key property is that we need to obtain both an arriving agent's data point z and her cost c. The reason is that the cost is used to importance-weight the data based on the probability of picking a price larger than that cost. (The cost report is also required by [Roth and Schoenebeck 2012] for the same reason.) As discussed in Section 2, a naïve implementation of this model is incentive-compatible but not strictly so. Exploring implementations, such as the trusted third party approach mentioned, is an interesting direction. For instance, in a strictly truthful implementation, the arriving agent can cryptographically commit to a bid, *e.g.* by submitting a cryptographic hash of her cost. Then the prices are posted by the mechanism. If the agent accepts, she reveals her data and her cost, verifying that the cost hashes to her commitment. It is strictly truthful for the agent to commit to her true cost.

This paper focused on the learning-theoretic aspects of the problem, but exploring the model further or proposing alternatives is also of interest for future work.

7.2. Conclusions and Directions

The contribution of this work was to propose an *active* scheme for learning and pricing data as it arrives online, held by strategic agents. The active approach allows learning from past data and selectively pricing future data. Our mechanisms interface with existing no-regret algorithms in an essentially black-box fashion (although the proof depends on the specific class of algorithms). The analysis relies on showing that they have good guarantees in a model of no-regret learning with purchased data. This no-regret setting may be of interest in future work, to either achieve good guarantees with no foreknowledge at all other than the maximum cost, or to propose variants on the model.

The no-regret analysis means our mechanisms are robust to adversarial input. But in nicer settings, one might hope to improve on the guarantees. One direction is to assume that costs are drawn according to a known marginal distribution (although the correlation with the data is unknown). A combination of our approach and the posted-price distributions of Roth and Schoenebeck [2012] may be fruitful here.

Broadly, the problem of purchasing data for learning has many potential models and directions for study. One motivating setting, closer to crowdsourcing, is an active problem where data points consist of pairs (example, label) and the mechanism can offer a price for anyone who obtains the label of a given example. In an online arrival scheme, such a mechanism could build on the importance-weighted active learning paradigm [Beygelzimer et al. 2009].

Acknowledgments

The authors thank Mike Ruberry for discussion and formulation of the problem. Thanks to the organizers and participants of the 2014 Indo-US Lectures Week in Machine Learning, Game Theory and Optimization, Bangalore.

We thank the support of the National Science Foundation under awards CCF-1301976 and IIS-1421391. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors alone.

REFERENCES

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 2009.

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In *Proceedings of the 23rd International Conference on Machine Learning (ICML-06)*, 2006.

Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine learning*, 80(2-3):111–139, 2010.

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *26th International Conference on Machine Learning (ICML-09)*, 2009.

Alina Beygelzimer, Daniel Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. In *Advances in Neural Information Processing Systems (NIPS-10)*, 2010.

Nicolo Cesa-Bianchi and Gabor Lugosi. Prediction, learning, and games. 2006.

Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *Information Theory, IEEE Transactions on*, 50(9): 2050–2057, 2004.

Rachel Cummings, Katrina Ligett, Aaron Roth, Zhiwei Steven Wu, and Juba Ziani. Accuracy for sale: Aggregating data with a variance constraint. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 317–324. ACM, 2015.

Ofer Dekel, Felix Fischer, and Ariel D Procaccia. Incentive compatible regression learning. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 884–893. Society for Industrial and Applied Mathematics, 2008.

Arpita Ghosh and Aaron Roth. Selling privacy at auction. In *Proceedings of the 12th ACM Conference on Electronic Commerce (EC-11)*, 2011.

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*, pages 931–948. ACM, 2014.

Steve Hanneke. *Theoretical foundations of active learning*. ProQuest, 2009.

Thibaut Horel, Stratis Ioannidis, and Muthu Muthukrishnan. Budget feasible mechanisms for experimental design. In *Latin American Theoretical Informatics (LATIN-14)*, 2014.

Katrina Ligett and Aaron Roth. Take it or leave it: Running a survey when privacy comes at a cost. In *The 8th Workshop on Internet and Network Economics (WINE-12)*, 2012.

Reshef Meir, Ariel D Procaccia, and Jeffrey S Rosenschein. Algorithms for strategyproof classification. *Artificial Intelligence*, 186:123–156, 2012.

Aaron Roth and Grant Schoenebeck. Conducting truthful surveys, cheaply. In *13th Conference on Electronic Commerce (EC-12)*, 2012.

Burr Settles. From theories to queries: Active learning in practice. *Active Learning and Experimental Design W*, pages 1–18, 2011.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 2012.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2000.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. 2003.