Delegating Data Collection in Decentralized Machine Learning

Nivasini Ananthakrishnan, Stephen Bates, Michael I. Jordan, and Nika Haghtalab

University of California, Berkeley

Abstract

Motivated by the emergence of decentralized machine learning (ML) ecosystems, we study the delegation of data collection. Taking the field of contract theory as our starting point, we design optimal and near-optimal contracts that deal with two fundamental information asymmetries that arise in decentralized ML: uncertainty in the assessment of model quality and uncertainty regarding the optimal performance of any model. We show that a principal can cope with such asymmetry via simple linear contracts that achieve $1 - 1/e$ fraction of the optimal utility. To address the lack of a priori knowledge regarding the optimal performance, we give a convex program that can adaptively and efficiently compute the optimal contract. We also study linear contracts and derive the optimal utility in the more complex setting of multiple interactions.

1 Introduction

The design of machine learning pipelines is increasingly a cooperative, distributed endeavor, in which the expertise needed for the design of various components of an overall pipeline is spread across many stakeholders. Such expertise pertains in part to classical design choices such as how much and what kind of data to use for training, how much test data to use for verification, how to train a model, and how to tune hyper-parameters, but, more broadly, expertise may reflect experience, access to certain resources, or knowledge of local conditions. To the extent that there is a central designer, their role may in large part be that of setting requirements, developing coordination mechanisms, and providing incentives.

Overall, we are seeing a flourishing new industry at the intersection of ML and operations which makes use of specialization and decentralization to achieve high performance and operational efficiency. Such an ML ecosystem creates a need for new design tools and insights that are not focused merely on how the designer could perform a task in this pipeline, but rather how she should delegate it to agents who are willing and capable of performing the task on her behalf. *How should the designer interact with this ecosystem? How should she evaluate and compensate other agents for their work? How does the outcome of the delegated pipeline compare with the outcome if the designer were to perform the task by herself?* In this work, we initiate the study of delegating machine learning pipelines through the lens of *contract theory* and take a step towards answering these questions.

Contract theory provides a principal-agent perspective, where the principal—who is the designer interested in the outcome of the learning pipeline—can create a contractual arrangement—a menu of services and compensations—with an agent. At the heart of the issue is creating contracts that incentivize the agents, who may be more knowledgeable and skilled than the principal, to take the appropriate actions. The uncertain and data-centric nature of machine learning tasks brings to light interesting sources of knowledge asymmetry between the principal and the agent and requires extensions of classical contract theory.

Consider a scenario where a firm delegates a predictive task to an ML service provider. In this context, the service provider may offer the firm either a dataset for learning or a pre-trained predictive model based on that dataset. To ensure aligned incentives, the firm needs to assess the dataset or predictive model and design the payment structure for the service provider accordingly. Since the accuracy of the model is crucial to the firm as it directly influences revenue, a natural evaluation approach involves directly measuring the accuracy of the model that the service provider produces for the firm. Several challenges arise during this evaluation process. Firstly, the firm generally only has limited data in the form of historical data or data acquired shortly after deploying the model for evaluation. So there is inherent noise in the evaluation. Secondly, the firm lacks knowledge about the baseline accuracy that is realistically achievable. This makes it harder for the firm to reward the service provider in a way that yields accuracy close to the optimal accuracy. These challenges are due to two sources of uncertainty and asymmetry that we study in this work:

- *Hidden actions* (aka Moral Hazard): Contracts must compensate the agent for his effort towards creating a good outcome for the principal. These contracts therefore must depend on observable and verifiable outcome quality, such as the true accuracy of a classifier. This is particularly challenging in machine learning pipelines, where the accuracy of the learned model is unknown a priori and random. The principal may be able to invest in resources, such as large test sets, that reduce this uncertainty at a cost and better evaluate the agent's effort. An important consideration here is determining whether the principal must accurately verify the outcome or instead incentivize the agent in the first place to ensure a high-quality outcome.
- *Hidden state* (aka Adverse Selection): Effective contracts use the knowledge of the best achievable outcome to incentivize the agents to work towards such outcomes. This is challenging in machine learning pipelines where the true error of the best model is unknown. Furthermore, generic methods that estimate the optimal accuracy tend to use almost as many resources as are needed to learn a model of that accuracy. Here again, we must ask whether contracts exist that appropriately incentivize the agent to perform his best while knowing very little about the optimal possible accuracy.

1.1 Our results

We consider performance-based contracts where the agent is compensated as a function of the estimated accuracy of the learned model. The principal's utility is the true accuracy of the learned classifier minus the monetary transfer she makes to the agent. We model the agent in two delegation settings. In each we contrast the principal's utility through contracting with and without information asymmetry. Borrowing terminology from the contract theory literature, we refer to the hypothetical scenario without information asymmetry as the *first-best scenario* and the resulting optimal utility as the *first-best utility*, which we use as a benchmark.

Single-round of interaction. We address both types of information asymmetry—hidden state and hidden action—creating contracts specific to each while also evaluating their efficacy when both asymmetries coexist. For hidden actions, our linear contract based on a single test point (Proposition 2) ensures at least $1 - 1/e$ fraction of the first-best utility. This guarantee continues to hold even with hidden state if the agent's sampling cost is low (Theorem 1).

For the hidden state challenge with n possible states, we derive an optimal contract by solving a convex optimization problem with $O(n^2)$ constraints. Section 4 describes how this contract's optimality improves as the principal's test set size increases.

Multiple rounds of interaction. In Section 5, we analyze a multi-round delegation setting where the agent is uncertain about the delegated task and uses feedback over rounds to learn the principal's requirements and collect relevant samples. We define the principal's regret and establish a tight $\Theta(T^{3/4})$ bound on this regret through repeated linear contracts over T rounds. This shows that linear contracts are also powerful approximations of optimal utility in multi-round settings. In comparison, we obtain a strictly better regret of $O(T^{2/3})$ for multi-round first-best contracts.

1.2 Related work

There is a rich literature on contract theory in economics (see, e.g., Laffont and Martimort, 2009; Bolton and Dewatripont, 2004). More recently, there has been work on algorithmic and statistical aspects of contract theory (Carroll, 2015; Dütting et al., 2019; Dütting et al., 2020; Bates et al., 2022; Alon et al., 2022) which include results on approximation by simple contracts. These results hold for either finite actions or outcomes, and thus are not directly applicable to our setting, which involves infinite actions and a continuous space of outcomes. Working in such spaces requires utilizing the structure of our problem, and specifically exploiting fundamental results on statistical minimax rates.

The *pricing* of data has been considered for various purposes and considerations (Bergemann and Bonatti, 2019; Acemoglu et al., 2022; Cai et al., 2015; Ho et al., 2016) including in learning problems (Agarwal et al., 2019; Chen et al., 2022). The latter study the pricing of previously collected data to incentivize the seller and buyer to be forthright about the valuation and quality of their data, respectively. We are interested instead in pricing for the purpose of incentivizing the data collecting agent to exert effort to collect data. Some of these papers also consider incentivizing high-quality data labelling by relying on multiple labellers who can be compared. We study delegation of learning in the setting of a single agent.

An adversarial perspective on the delegation problem has been considered for machine learning from the lens of interactive proofs. In this line of work (Goldwasser et al., 2021; Chiesa and Gur, 2018), the principal wants to fully verify the effort of an agent who may be an adversary that is interested in getting his effort verified. While they deal with similar challenges, such as not knowing the optimal achievable error, they do not consider incentivizing the agent (via contracts and compensations) to improve the outcome.

Concurrent work by Saig et al. (2023) studies a similar setting of incentivizing data collection for classification. They characterize the optimal contract for a given test set size, under the hidden action challenge, as a threshold contract when the agent has two choices for actions. They provide conditions which make the threshold contract optimal even for additional actions. They study empirically the effect of the hidden state challenge. We provide results for an arbitrary number of actions and propose a contract that is based on a single test sample that is optimal relative to what is achievable without the hidden action and hidden state challenges. We show that this contract is robust to hidden state challenges in many cases and describe other approaches of dealing with hidden state challenges outside of these cases.

2 Model

We have a task distribution $\mathcal{D}$ representing the joint distribution over the domain and label set. The principal aims to learn a classifier h that achieves high accuracy on $\mathcal{D}$, denoted by $1 - L_{\mathcal{D}}(h)$. To accomplish this, the principal delegates the task to an agent who selects the number of samples to collect and trains a classifier. We prioritize the collection of samples as the primary effort, considering it more significant than classifier training. The principal's primary objective is to incentivize high-quality data collection, leading to the development of an accurate classifier. To

evaluate the performance of the model obtained through delegation, the principal possesses an independent test set consisting of independently and identically distributed (i.i.d.) points drawn from the distribution $\mathcal{D}$. The principal utilizes this test set to evaluate the learned classifier's accuracy.

As in many other delegation settings, the principal faces the hidden state and hidden action challenges when delegating learning. While the principal desires to construct a contract based on the true accuracy of the learned model, $1 - L_{\mathcal{D}}(h)$, they can only obtain a noisy estimate of this value using test data. Our focus is on scenarios where the size of the test dataset is not excessively large. If the test dataset is too large, it becomes more beneficial for the principal to learn a model using their own test data rather than delegating the data collection process. Even when the estimate of the learned model's accuracy has negligible noise, the principal still faces the hidden state challenge, i.e., the principal does not know how to value the accuracy since she does not know the optimal error achievable. We use $1 - \theta$ to indicate the optimal accuracy achievable on $\mathcal{D}$. Assigning a low payment for the model's accuracy when the optimal error, θ, is high would result in negative agent utility, discouraging agent participation. Conversely, assigning a high payment when the optimal error is low might incentivize the agent to collect a smaller dataset than is optimal for the principal.

The delegation process begins with the principal publishing a contract which is a mapping from test accuracy to payment for the agent. Seeing the contract, the agent collects data and provides a classifier to the principal. The principal then executes the contract by evaluating the classifier on her own test set. The principal pays the agent the amount specified by the contract for the measured test accuracy. We assume that the principal can commit to a test set in advance and that this test set is not accessible to the agent until the contract is executed after the agent's data collection.

Utilities. Upon receiving a classifier with accuracy a for the task distribution $\mathcal{D}$ and paying the agent t, the principal gets utility $a - \beta t$ for some constant $\beta > 0$. The agent exerts effort α per sample it collects. So the utility for the agent receiving payment t by collecting n samples is $t - \alpha n$.

Outcome as a function of the agent's action. We assume that when the agent collects n samples,[1] the classifier's observed accuracy on the principal's test set drawn from $\mathcal{D}$ is drawn from a distribution with mean $1 - \theta - \frac{d}{n^p}$ and variance that is determined by the size of the test set. The constant d depends on the complexity of the training algorithm and the constant p describes the rate of decay of the excess error. These rates are motivated in part by minimax statistical rates and scaling laws.

Even though minimax rates are typically upper bounds, we treat them as exact rates in the main body and defer the discussion on the implications of treating them as upper bounds to Appendix B.1.

Remark 1 (VC dimension bound)**.** *An algorithm that PAC-learns a function class $\mathcal{H}$ with VC dimension d using n i.i.d. samples drawn from $\mathcal{D}$ and returns a classifier h satisfying $L_{\mathcal{D}}(h) \leq \theta(\mathcal{D}, \mathcal{H}) + C\sqrt{d/n}$, where $\theta(\mathcal{D}, \mathcal{H}) = \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$. This is minimax-optimal as there is a distribution $\mathcal{D}$ such that $L_{\mathcal{D}}(h) \geq \theta(\mathcal{D}, \mathcal{H}) + C\sqrt{d/n}$.*

Remark 2 (Linear regression model)**.** *In a d-dimensional linear model with covariates $x_i \sim \mathcal{N}(0, \Sigma)$ and outcomes $y_i = \beta^t x_i + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ for $i \in [n]$, the Ordinary Least Squares (OLS) estimator $\hat{\beta}$ satisfies the property $\mathbb{E}[(x^t \hat{\beta} - y_i)^2] = \sigma^2 (1 + O(d/n))$.*

First-best contracts. As a benchmark for the best performance we can hope to achieve, we first consider the problem in an idealized setting without the hidden state and hidden action challenges. This is when the principal knows the optimal error θ and the mapping between the agent's action n and the test accuracy of the resulting model is deterministic (i.e., is exactly $1 - \theta - \frac{d}{n^p}$). The optimal contract in this idealized setting is called the *first-best contract*. The next proposition provides a closed form for this contract.

[1] We will consider agent's action as continuous and the true sample size is a rounding of the action.

Proposition 1 (First-best contract)**.** *For any set of problem parameters $\theta \in [0, 1), d, p, \alpha, \beta > 0$, the first-best contract offers payment αn^* when the test accuracy is at least $1 - \theta - d/n^{*p}$, where $n^* = (pd/\alpha\beta)^{1/(p+1)}$.*

One way to interpret the first-best contract is that it asks the agent to collect n^* samples and compensates the agent exactly for n^* samples. Without hidden state or hidden action, the first-best contract yields zero utility to the agent. In this idealized scenario, the principal's utility due to the first-best contract is called the *first-best utility* and serves as a benchmark for comparison in our analysis of delegation. While first-best utility is used as a benchmark, the first-best contract itself may not be optimal due to existing randomness in test accuracy (hidden action). Additionally, each optimal error value θ leads to a different first-best contract, which is not implementable when the principal doesn't know the θ parameter exactly (hidden state). When dealing only with hidden action but known θ, the principal's goal is to set up a contract specified for θ that deals with the randomness in the test accuracy to recover some fraction of the first-best utility. However, when both actions and states are unknown (uncertainty in θ and test accuracy) the contract must ensure good principal utility for a range of possible states θ.

Linear contracts. As opposed to first-best contracts that can be quite complex, *linear contracts* are simple contracts that compensate an agent by a linear function of the test accuracy. That is, a c-linear contract for parameter $c \in \mathbb{R}^+$ assigns payment $T_c(a) = c \times a$ when the test accuracy is a.

Linear contracts must have non-negative parameter c, since the principal cannot make negative payments to the agent.

3 Optimality of Linear Contracts

In this section, we aim to find near-optimal contracts in the realistic scenario with hidden state and hidden actions, recognizing that the first-best contract may not be optimal. Our main result is that a linear contract compensating the agent based on the test (and not true) accuracy is approximately optimal across all possible contracts for the principal. Moreover, the slope of the linear contract has an explicit value that is the same across a wide range of θ making it possible to deal with both hidden state and hidden state challenges.

A crucial advantage of our linear contract is that it works with any unbiased estimator of the accuracy of the learned model. Therefore, even a test set of size one suffices to enact this contract. We state our main results in this section and defer their formal proofs to Appendix A.

Consider the hidden action (but known state) challenge where the principal knows θ but not the random mapping from the agent's action n to test accuracy. This mapping has a mean $1 - \theta - \frac{d}{n^p}$) and a variance dependent on the test set size. The variance is finite but possibly arbitrarily large. This setting includes the delegation problem where the principal has as little as just a single sample $x \sim \mathcal{D}$ in her test set. Furthermore, we assume that beyond knowing the mean of the distribution of the test accuracy $1 - \theta - \frac{d}{n^p}$, the distribution can be arbitrary and unknown to the principal.

Our main result is that we can design an approximately optimal linear contract. Furthermore, under a wide range of problem parameters θ, the principal does not even need to know the optimal error to construct this contract. This allows us to deal with both hidden state and hidden action challenges. Our results in fact show a stronger comparison, that linear contracts approximate not just the optimal utility but also the first-best utility. This is quite a strong guarantee as there is often no contract that can achieve the first-best utility in presence of the hidden action challenge.

Before we state our main theorem, we start with the following proposition which deals only with the hidden action challenge while assuming that optimal error θ is known to the principal. Our

main result in Theorem 1 follows from this proposition and shows that the linear contract in this proposition is also a good choice in more general settings.

Proposition 2 (Linear contracts are approximately optimal when optimal error is known)**.** *For any set of problem parameters $\theta \in [0,1), d, p, \alpha, \beta > 0$, if the principal knows θ (but not the distribution of the test error) she can construct a linear contract that brings an expected utility that is at least $1 - 1/e$ times the first-best utility. Furthermore this contract only requires a single test sample.*

The linear contract c^* that achieves this approximately optimal utility is the following:

$$c^* = \max\left(\frac{1}{\beta(p+1)^{\frac{p+1}{p}}}, \frac{\alpha d^{\frac{1}{p}}}{p} \cdot \left(\frac{p+1}{1-\theta}\right)^{\frac{p+1}{p}}\right).$$

At a finer level, this linear contract approximates the first-best utility by a factor of

$$1 - \frac{1}{(p+1)^{\frac{p+1}{p}}} \geq 1 - \frac{1}{e}.$$

Let us first note that previous work (Alon et al., 2022; Dütting et al., 2019) has provided constant approximation guarantees but is limited to settings where the agent's action set is a finite set or where the ratio of the maximum and minimum reward for the principal is bounded by H. In the former case, an approximation ratio of $1/2$ is obtained and in the latter the ratio is $1/2\log(H)$. Neither of these conditions hold in our settings, as the action is the number of samples collected and is unbounded and the reward can take any value in $(0,1)$. Instead, we use the structure of first-best contract (Proposition 1), the linearity of contracts, and the structure of the utility functions to obtain this $1 - 1/e$ approximation guarantees.

Proof sketch of Proposition 2. The full details are deferred to the appendices; here we provide some intuition and a proof sketch. Underlying the proof is the linearity of expectation and the fact that the agent is expectation-maximizing. Under a linear contract c, the expectation-maximizing agent aims to maximize $\mathbb{E}[c \cdot a(n) - \alpha n] = c \cdot \mathbb{E}[a(n)] - \alpha n$, where $a(n)$ is the test-set accuracy of a model trained on n samples drawn from an unknown distribution with mean $1 - \theta - d/n^p$. The only distribution-dependent quantity in this maximizing objective is the expected accuracy $\mathbb{E}[a(n)] = 1 - \theta - d/n^p$. So the agent's action and hence the principal's contract design only depends on the expectation of the test accuracy and not on the exact distribution of the test accuracy. Next we sketch a proof for the approximation result and use the structured way the expected accuracy depends on the number of samples drawn.

Note that c^* is the maximum of two terms. Let us denote these terms by c_1, c_2. Given a linear contract with parameter c, the agent's best response is to choose n so as to maximize $u(n; c) = c\,(1 - \theta - d/n^p) - \alpha n$. The maximizing value is $n(c) = (cdp/\alpha)^{\frac{1}{p+1}}$. By setting c large enough, we have $u(n(c), c) \geq 0$ where c_2 is the threshold above which this holds. So the value of c_2 is set to ensure the agent gets non-negative utility from participating.

When $c_1 \geq c_2$, c_1 satisfies the participation constraint. By computing the principal's utility from the linear contract c_1 using the expression for the agent's best response, we see that it is $1 - \beta c_1$ times the first-best utility. Moreover, we have $1 - \beta c_1 = 1 - 1/(p+1)^{\frac{p+1}{p}}$. It turns out the same upper bound holds for the approximation ratio of the linear contract c_2 to the first-best utility when $c_2 \geq c_1$. This upper bound is decreasing in p and the limit as $p \to 0$ is $1 - 1/e$. □

Importantly, by inspecting the contract in Proposition 2, we see that in many cases it does not depend on problem-specific parameters like the optimum error. This makes c^* deployable in practice.

The optimal-error-parameter-agnostic linear contract is appropriate when the cost per sample collection is small enough and when the optimal error is low enough. As a result, when α is small, we can relax the assumption that the principal knows the exact optimum error θ to that the principal knows that θ lies in a certain range. Moreover, even under this relaxation, linear contracts are still approximately optimal. This is stated as the following theorem.

Theorem 1 (Main result)**.** *For any $d, p, \beta > 0$, consider the linear contract $\bar{c} = \frac{1}{\beta(p+1)^{\frac{p+1}{p}}}$. For any $\overline{\theta} \in [0, 1)$, suppose the optimum error θ is any value in $[0, \overline{\theta})$ and that $0 < \alpha \leq \frac{p}{\beta d^{1/p}} \left(\frac{1-\overline{\theta}}{(p+1)^2}\right)^{\frac{p+1}{p}}$. Then, $\bar{c}$ has utility at least $(1 - 1/e)$ times the first-best utility.*

Note that $\bar{c}$ is constructed based on p (error decay rate) and β (how the principal values accuracy relative to payment). The principal knows these quantities. In contrast, the optimal contract requires additional knowledge, such as θ (optimum error) and α (agent's cost per sample). The theorem demonstrates a simple contract that requires less knowledge but remains approximately optimal in utility.

4 Extensions

Medium test set regime. In our analysis, we have examined the impact of the hidden action challenge when dealing with a small test set size. The significance of the hidden action challenge diminishes as the test set size increases, as the principal can obtain highly accurate estimates of the model's accuracy. However, when the test set becomes too large, delegation loses its value since the principal can independently learn an accurate model without delegation. Is there a regime in which the test set size is large enough for hidden action to not be significant while also being small enough for the principal to benefit from delegating data collection? In this section, we demonstrate the existence of such a regime, referred to as the "medium test set regime." Later, we outline how we can capitalize on the larger size of the test set to achieve stronger results.

The sample complexity for learning an ϵ-optimal model is $\Theta(d/\epsilon^2)$. In particular, this bound is linear in the training algorithm's complexity which can be problematic when using highly complex training algorithms. We say that the medium test set regime exists, if the sample complexity for hidden action is significantly smaller than $\Theta(d/\epsilon^2)$, where ϵ captures the significance level of hidden action which we will make precise in the following definition.

Definition 1 (Insignificance of hidden action at level ϵ)**.** *In a finite test set setting with hidden action, for any optimal error parameter θ, let* OPT *denote the optimal expected utility of contracting. We say that hidden action is insignificant at level ϵ, for any $\epsilon > 0$, if the expected utility of the first-best contract based on θ in this setting is at least* $\text{OPT} - \epsilon$.

We next state a theorem giving the sample complexity of the principal's test set to achieve insignificance of the hidden action. The sample complexity stated in the theorem is logarithmic in d while learning would have required a number of samples linear in d. This demonstrates the existence of a medium test set regime where it is possible to employ delegation without considering hidden action.

Theorem 2 (Sample complexity for insignificant hidden action)**.** *For any $\epsilon > 0$, if the principal has a test set of size $O\left(\frac{1}{\epsilon^2} \log \frac{d}{\epsilon}\right)$, then hidden action is insignificant at the level ϵ.*

4.1 Optimal contracts for hidden state

By ignoring hidden action in the medium test set regime, we can hope to design contracts with stronger guarantees. Previously we were able to design contracts with high utility when the optimal error lies in a particular range given in Theorem 1. By ignoring hidden action, we can design contracts with utility guarantees for when the optimal error lies in any arbitrary set. When the principal holds a finitely supported prior belief over the optimal error value, we show how to compute the optimal contract by setting up a convex optimization problem. We also describe some qualitative properties of the optimal contract in this setting.

When we ignore the hidden action challenge, we can assume that the observed accuracy is deterministic in the agent's action. That is, when the agent collects n samples, the observed accuracy is $a(n, \theta) = 1 - \theta - d/n^p$. We assume that the principal holds a prior belief on the optimal error but does not know the exact value. The agent knows more about the optimal error since he collects data that informs him more about the optimal error. We assume that the agent knows the exact optimal error. We start by analyzing the optimal contract in this setting. Later in Section 4.2, we discuss how to design contracts in the more realistic setting of the agent learning the optimal error instead of knowing this value exactly. And we show that the utility guarantees by making the perfectly aware agent assumption still hold approximately in the more realistic case with a learning agent.

Let us analyze the optimization problem for computing the optimal contract. Let the finite support of the prior over optimal error be $\{\theta_1, \ldots, \theta_N\}$. The principal puts forth a contract of accuracy-payment pairs $\{(a_i, t_i) : i \in [N]\}$ with the pair i intended for when the optimal error is θ_i.[2] Let us denote the expected accuracy from collecting n_i when optimal error is θ_i by $a_i = a(n_i, \theta_i)$. Here n_i is the number of samples the agent would collect to achieve accuracy a_i when optimal error is θ_i. The principal optimizes over $(n_i, t_i)_{i\in[N]}$. The constraints of the optimization problem for the principal's contract design for hidden state are one of two types. The first type of constraint is the participation constraint, which ensures that the agent is adequately compensated for his effort when he chooses the contract intended for the optimal error. For each $i \in [N]$, the participation constraint (PC_i) can be expressed as $\alpha n_i \le t_i$, where α represents the compensation rate.

The second type of constraint is the incentive compatibility constraint to ensure that the agent chooses the option intended in the contract for the optimal error. For any $i, j \in [N]$, the corresponding incentive compatibility constraint is that when the optimal error is θ_i, the utility of choosing (a_j, t_j) is worse for the agent than choosing (a_i, t_i). The number of samples the agent would choose to achieve a_j accuracy under optimal error θ_i is n_{ij} such that $a_j = a(n_{ij}, \theta_i)$.[3] The constraint ($\text{IC}_{ij}$) is $t_j - \alpha n_{ij} \le t_i - \alpha n_i$. Due to the structure of $a(n, \theta)$, the IC constraints are convex. The principal's expected utility which it maximizes is $\sum_{i=1}^{N} \nu(\theta_i)(a_i - \beta t_i)$. So the contract design problem is the following optimization problem:

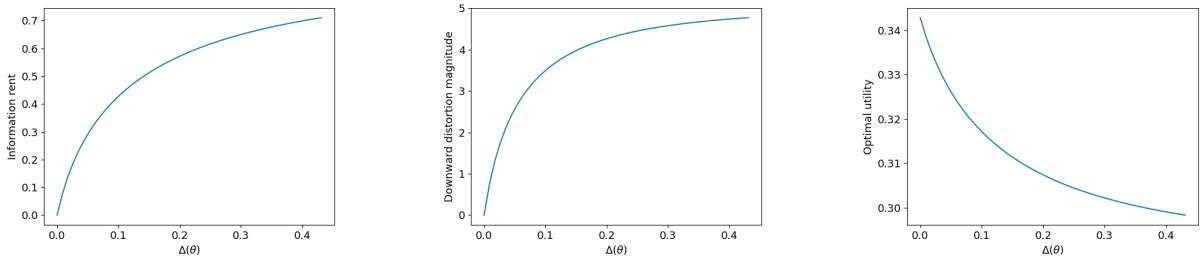
$$\begin{aligned} \min_{(n_i,t_i)_{i=1}^N} \quad & \sum_{i=1}^{N} \nu(\theta_i)(a_i - \beta t_i) \\ \text{s.t.} \quad & \alpha n_i \le t_i, \quad i \in [N] \\ & t_j - \alpha n_{ij} \le t_i - \alpha n_i, \quad i, j \in [N] \\ & n_i, t_i \ge 0, \quad i \in [N]. \end{aligned} \tag{Opt}$$

[2]This is implied by the revelation principle that states that, with hidden state, any delegation mechanism is equivalent to an *incentive compatible* mechanism where all agents inform their private information to a planner who then recommends actions.

[3]Note that all accuracies cannot be achieved for all optimal errors. If no such n_{ij} exists, an incentive compatibility constraint is not needed.

Qualitative insights on the optimal contract. We derive the following insights (see Figure 4.1) when there are two values for the optimal error, $\theta_1 < \theta_2$, in the Appendix B.3. These properties also hold more generally for finitely supported beliefs and have been studied for classical contract design for many other delegation problems Laffont and Martimort (2009).

- *Decreased utility*. The principal gets lower utility than the first-best utility and this utility decreases as $\Delta\theta = \theta_2 - \theta_1$ increases.
- *Information rent.* In the first-best contract, the agent gets no more payment than to compensate his effort. That is, $t = \alpha n$. Under hidden state, for problems with lower optimal error, the agent gets positive utility. This information rent is to incentivize the agent to not pretend the problem is harder and exert lower effort to achieve an accuracy that requires more effort if the problem was harder.
- *Downward distortion.* The first-best contract calls for the agent to collect a particular number of samples regardless of the optimal error. Under hidden state, when the problem is harder, agents are asked to collect fewer samples compared to the first-best contract. When the problem is the easiest in the support, the agent is asked to collect the same number of samples as in the first contract.



(a) Information rent vs $\Delta\theta$ (b) Downward distortion magnitude vs agent's $\Delta\theta$ (c) Optimal utility vs $\Delta\theta$

Figure 1: Variation of utility, information rent and downward distortion magnitude with the gap in optimal error values ($\Delta\theta$). Information rent is the utility the agent makes under the lower optimal error problem. Downward distortion magnitude is how many fewer samples the agent collects compared to the first-best contract under the higher optimal error problem.

4.2 Designing contracts against state-learning agents

In Section 4.1 and particularly Opt, we assumed perfect knowledge of the hidden state (θ) by the agent. However, in reality, the agent does not know the optimal error beforehand. Instead, as the agent executes the contract, he learns more about the optimal error and adapts his actions accordingly. To design a contract for such a state-learning agent, the principal would need to predict the agent's response to the contract. However, this is challenging for arbitrary contracts since the principal would require knowledge of the agent's exact learning strategy, which is often unreasonable. Therefore, we focus on analyzing simple contracts for which we can easily derive the agent's response. We demonstrate numerically that the utility achieved with these simple contracts is close to the utility we previously derived for state-aware agents, which we refer to as "state-aware utility." This provides evidence that qualitative insights we derived about the state-aware utility in Section 2 and Section 3 are applicable in the more realistic case of a state-learning agent. We focus on the case

where the optimal error can take one of two possible values, $\theta_1 < \theta_2$, but these design principles also extend to more possible values of the optimal error. The simple contract we consider, which we call the *state-learning contract*, is the best of two simple contracts: optimal *pooling* and *separating* contracts.

Separating contract. Separating contracts allow the agent to perfectly infer the hidden state while executing the contract. These are incentive-compatible contracts that ask agents to collect n_1, n_2 samples under optimal errors $\theta_1 < \theta_2$ respectively. Additionally, n_1, n_2 are such that the agent can successfully infer the optimal error after collecting $\min(n_1, n_2)$ samples. The agent's response to this contract would be to first collect $\min(n_1, n_2)$ samples and decide whether to collect more depending on the inferred optimal error. The agent's successful inference of the optimal error makes computing optimal separating contracts similar to the contract design problem against a state-aware agent, which was solving the optimization problem Opt. The new optimization problem that yields optimal separating contracts has the same objective and constraints as Opt with the added constraint that $n_1, n_2 > n_0$ for some n_0 that we will describe soon. The additional constraint ensures that the agent knows the optimal error (with high probability) after collecting $\min(n_1, n_2)$ samples.

To determine the value of n_0, we rely on assumptions about the agent's learning strategy. We assume that the agent can distinguish between θ_1, θ_2 with high probability using $k/(\Delta\theta)^2$ samples. Here $\Delta\theta = \theta_2 - \theta_1$ and k is a constant reflecting the degree of assumption made about the agent's efficiency. A lower value of k is a stronger assumption, assuming a more efficient agent. This assumption on the agent's learning strategy is more reasonable compared to assuming precise knowledge of the agent's learning strategy.

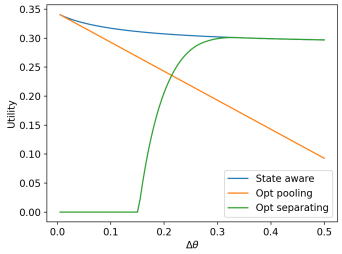
When n_0 is small enough that the added constraint $n_1, n_2 > n_0$ is not active, the state-learning agent is behaving exactly as the state-aware agent, so our results from Section 4.1 apply. On the other hand, if n_0 is large (which happens when $\Delta\theta$ is small) the additional constraint becomes too restrictive and the utility becomes low. In this case, another approach works well.

Pooling contract. In pooling contracts, the agent has no incentive to learn the optimal error. The pooling contract asks the agent to achieve one accuracy level $\bar{a}$ regardless of the optimal error. The payment for this accuracy is set to ensure the agent can get nonnegative utility regardless of the optimal error. It is again straightforward to understand the agent's response to this contract. Suppose $\bar{a}$ can be achieved by collecting $\bar{n}_1 < \bar{n}_2$ samples under optimal errors $\theta_1 < \theta_2$ respectively. To execute this contract, the agent starts collecting $\bar{n}_1$ and sees if it achieves $\bar{a}$ accuracy. If it does not, he collects $\bar{n}_2 - \bar{n}_1$ more samples since this action is guaranteed to yield nonnegative utility. Furthermore, collecting fewer or no additional samples results in less than $\bar{a}$ expected accuracy and hence zero payment even though the agent exerted effort.

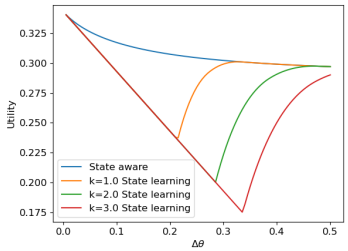
A pooling contract does not let agents differentiate actions for different optimal errors and would be sub-optimal for this reason. However, when the difference in both problems is not significant i.e., $\Delta\theta$ is low, the benefit to the principal for distinguishing the agents is low. In summary, the separating contract has good utility when $\Delta\theta$ is large and the pooling contract has good utility when $\Delta\theta$ is small. By deploying the contract of the two with the higher utility, we can hope to have good utility for all values of $\Delta\theta \in [0, 0.5]$.

Numerical results. We compute the utility difference between the state-aware contract and the state-learning contract, varying problem parameters $\Delta\theta = \theta_2 - \theta_1$ and k. We highlight a few observations (see Fig 2), that reflect the intuition we used to design the approach for state-learning contracts.

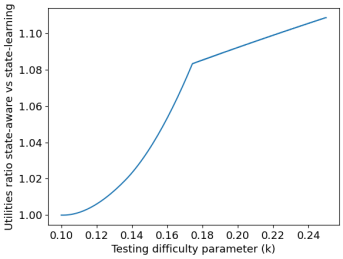
- Figure 2a shows that the state-learning contract is pooling when $\Delta\theta$ is less than some threshold and is separating otherwise. For small and large values of $\Delta\theta$, the state-learning contract has utility close to the state-aware utility.



(a) Utilities of different contracts vs $\Delta\theta$



(b) Utilities of state-aware and state-learning contracts for various k's vs $\Delta\theta$



(c) Worst-case sub-optimality of state-learning contracts relative to state-aware contracts

Figure 2: Figure 2a plots the utilities of the state-aware, separating, and the pooling contract against ($\Delta\theta$). Figure 2b again plots the utilities of contracts on the y-axis and $\Delta\theta$ on the x-axis. It plots the state-aware utility and the utilities of state-learning contracts of different levels k of agent's testing efficiency. Figure 2c plots the worst-case sub-optimality of state-learning contracts against k. The sub-optimality is the ratio of the state-learning contract's utility and the state-aware utility. The worst-case sub-optimality is the largest sub-optimality over all $\Delta\theta \in [0, 0.5]$.

- Figure 2b shows that when it is more difficult to distinguish between θ_1, θ_2, the pooling contract is better than the separating contract for more values of $\Delta\theta$.
- Figure 2c shows that the worst-case sub-optimality over all $\Delta\theta$ values of the state-learning contract compared to the state-aware utility increases as k increases. When the agent can test more efficiently, the state-learning contract has greater utility for the principal.

5 Multi-round Delegation

So far, we analyzed delegated learning that occurs through a single round of interaction between the principal and the agent. However, delegation often occurs over multiple rounds to allow the agent to learn more about the principal's requirements. Here we model such a scenario and analyze what happens when the principal uses a linear contract in each round. We introduce a notion of regret and show that repeated linear contracting over T rounds results in $\Theta(T^{3/4})$ regret for the principal which is worse than the $O(T^{2/3})$ regret achievable without delegation i.e., the *first-best regret*. We provide proof sketches of our results in the main body and provide the full proofs in the appendices.

The model. To model uncertainty about the principal's requirements, we assume that the target distribution D^* belongs to a class $\mathcal{D} = \{D_1, \ldots, D_k\}$. The agent knows the class $\mathcal{D}$ but does not know D^* apriori. The principal contracts and deploys classifiers for T rounds.

Contracting Protocol. For each round $i = 1, \ldots, T$:

1. The principal announces payment rule ρ_i which is a randomized mapping from a classifier to a positive, real-valued payment to the agent.
2. The agent chooses a target number of i.i.d. samples to collect from each distribution. Denote this number by $\mathbf{n_i} = (n_i^{(1)}, \ldots, n_i^{(k)})$ where $n_i^{(j)}$ is the number of samples the agent draws from distribution D_j in round i. This choice is not observed by the principal.
3. The agent provides classifier h_i to the principal.

4. The principal deploys a classifier $\bar{h}_i$.

5. The principal pays $\rho_i(h_i)$ to the agent according to the announced payment rule ρ_i.

The principal's and agent's action in each round can be chosen adaptively depending on the actions and outcomes of previous rounds.

Utilities. Over these T rounds, the agent's utility is the sum of payments minus sample collection costs: $\sum_{i=1}^{T}\left(\rho_i(h_i) - \alpha\sum_{j=1}^{k} n_i^{(k)}\right)$. The principal's utility is the sum of accuracies minus payments: $\sum_{i=1}^{T}\left(1 - L_{D^*}(\bar{h}_i) - \beta\rho_i(h_i)\right)$.

Test-accuracy-based payments. Our results deal with contracts based on test accuracy of the deployed classifier. In round $i \in [T]$ where the principal deploys the classifier $\bar{h}_i$, the payment is a function of the test accuracy $1 - L_{D^*}(\bar{h}_i) + \eta_i$, where η_i is a mean-zero, random variable resulting in the principal's randomness of testing. A linear contract c_i in this round offers payment c_i times this test accuracy in this round.

Feedback-providing contracts. In these repeated interactions, the payments serve as feedback to the agent to learn the principal's requirements as long as the principal's deployed classifiers depend on the classifiers provided by the agent. We focus on *feedback-providing* contracts which satisfy the following property. A contract is feedback-providing if for every round t with $c_t > 0$, the principal deploys the agent-provided classifier, and the payment for the round is therefore c_t times the test accuracy of the agent-provided classifier.

We prove positive results on the utility the principal can achieve when contracting with a rational agent.

$\mathcal{H}$-regret. We introduce a notion of regret for the principal and the agent in this online setting. The regret notion is defined relative to a class of classifiers $\mathcal{H}$. It compares utility to the utility obtained by deploying the best classifier in $\mathcal{H}$ in every round, without any sample collection or payments.

Definition 2 ($\mathcal{H}$-regret)**.** *Let $\mathcal{H}$ be any class of classifiers. Let $((\mathbf{n}_t, h_t, \rho_t(h_t))_{t=1}^{T}$ be the sequence of actions by the principal and agent. The principal's $\mathcal{H}$-regret ($R_T^P(\mathcal{H})$) is the difference in the utility of the sequence and the utility of deploying the most accurate classifier in $\mathcal{H}$ without payments:*

$$R_T^P(\mathcal{H}) = T\max_{h\in\mathcal{H}}(1 - L_{D^*}(h)) - \sum_{t=1}^{T}\left(1 - L_{D^*}(h_t) - \mathbb{E}\left[\rho_t(h_t)\right]\right)$$

The agent's $\mathcal{H}$-regret ($R_T^A(\mathcal{H})$) is the difference in the utility of the sequence and the utility of deploying the highest expected payment yielding classifier in $\mathcal{H}$ and collecting no samples:

$$R_T^A(\mathcal{H}) = \sum_{t=1}^{T}\max_{h\in\mathcal{H}}\mathbb{E}\left[\rho_t(h)\right] - \sum_{t=1}^{T}\left(\rho_t(h_t) - \alpha\sum_{j=1}^{k} n_t^{(j)}\right).$$

We consider a model of rationality for the agent in repeated linear contracting. This is an assumption on the agent's $\mathcal{H}$-Regret rate.

Agent's rationality. Let $(c_1)_{t=t}^T$ be the sequence of linear contracts employed by the principal. A rational agent achieves $\mathcal{H}$-regret sub-linear in $\sum_{t=1}^T c_t$.

We discuss the implications of contracting with a rational agent first and then justify the rationality assumption by providing an algorithm for the agent to achieve the rationality criteria.

In the following proposition, we show how contracting with such a rational agent provides the principal a sub-linear $\mathcal{H}$-regret. The proof of the proposition constructs a contract that the principal can use to achieve sub-linear $\mathcal{H}$-regret. This contract makes use of an upper bound on the agent's regret rate.

Proposition 3. *Suppose the agent achieves $O\left(\left(\sum_{t=1}^T c_t\right)^x\right)$ agent's $\mathcal{H}$-regret with $x < 1$, for any sequence of contracts $(c_t)_{t=1}^T$. Then, the principal can achieve $O\left(T^{\frac{1}{2-x}}\right)$ $\mathcal{H}$-regret.*

Proof. The contract that allows this regret for the principal is described here. The principal contracts with $c_t = 1$ for the first $N = T^{1/(2-x)}$ rounds. In these rounds $t \in [1, N]$, the principal deploys the agent-provided classifier h_t. The principal shuts down contracting for the remaining rounds. That is, $c_t = 0$ for $t \in (N, T]$. In rounds $t \in (N, T]$, the principal deploys a classifier $\bar{h}$ that is uniformly, at random picked from the classifiers $h_1, \ldots, h_N$ deployed in the first N rounds.

In the first N rounds, the principal's $\mathcal{H}$-regret is bounded by $O(N)$ trivially. In the remaining $T - N$ rounds, the principal's $\mathcal{H}$-regret is bounded above by just the suboptimality of the deployed classifiers since no payments are offered. The regret in the last $T - N$ rounds is at most

$$\begin{aligned}(T-n)\left(\mathbb{E}[L_{D^*}(\bar{h})] - \theta^*\right) &= (T-n)\frac{1}{N}\sum_{i\in[N]}\left(\mathbb{E}[L_{D^*}(\bar{h})] - \theta^*\right) \\ &\leq (T-n)\frac{1}{N}\cdot \text{Agent's } \mathcal{H}\text{-regret} \\ &\in O\left(\frac{T}{N^{1-x}}\right),\end{aligned}$$

where the last step follows from the rationality assumption that the agent achieves a $\mathcal{H}$-regret of $O(N^x)$. The cumulative principal's $\mathcal{H}$-regret overall all T rounds is therefore $O(N + T/N^{1-x})$. Since $N = T^{1/(2-x)}$, principal's $\mathcal{H}$-regret is $O(T^{1/(2-x)})$.

□

The above proposition shows how the principal can achieve sublinear $\mathcal{H}$-regret under the agent's rationality assumption. Now we show that the agent's rationality can be achieved. In our proof, we provide an algorithm for the agent to achieve $\mathcal{H}$-regret sublinear in $\left(\sum_{t\in[T]} c_t\right)^{2/3}$.

Proposition 4. *For any sequence of feedback-providing contracts $(c_t)_{t=1}^T$, the agent can achieve $O\left(\left(\sum_{t\in[T]} c_t\right)^{2/3}\right)$ $\mathcal{H}$-regret.*

Proof sketch. We will describe the agent's algorithm here and defer the full analysis of this algorithm to the appendices. For any round t, let us denote $\sum_{t'\leq t} c_t$ by s_t. Since the agent incurs cost linear in the number of samples collected, the algorithm must limit the number of samples to $O\left(s_T^{2/3}\right)$. To do this, at round t, the agents collects $\max\left(0, \left\lfloor s_t^{2/3}\right\rfloor - \left\lfloor s_{t-1}^{2/3}\right\rfloor\right)$ samples. As a result, for any t, the number of samples collected up to round t is $\lfloor s_t^{2/3}\rfloor$.To select the distribution to sample from, the agent picks one out of $\{D_1, \ldots, D_k\}$ uniformly at random. This fully describes how the agent's algorithm collects samples by describing the number of samples and distributions to sample from.

The remaining component of the agent's algorithm is classifier selection. The classifier selection includes exploration and exploitation where the exploration is to determine which of $D_1, \ldots, D_k$ provides samples to train an accurate classifier. Since we know D^* is one of these k distributions, we are guaranteed that there exists a distribution providing relevant samples.

We assign rounds of sample collection to be rounds of exploration. These rounds are suitable for exploration for two reasons. The first is because the principal deploys the agent-provided classifiers in these rounds. By the algorithm's design, the agent only collects samples in rounds with $c_t > 0$, and in feedback-providing contracts, agent-provided classifiers are deployed in these rounds. Secondly, this choice also leads to the right number of exploration rounds for the optimal exploration-exploitation tradeoff. Let $\{i_1, \ldots, i_r\}$ denote the indices of rounds of exploration. This set's size is at most $\left(\sum_{t\in[T]} c_t\right)^{2/3}$. A phased exploration is done in these rounds.

The phased exploration divides the indices $\{i_1, \ldots, i_r\}$ into phases where phase j is of length 2^{j-1}. So at least 2^{j-1} samples are collected in phase j. The number of phases is therefore $\log\left(\left(\sum_{t\in[T]} c_t\right)^{2/3}\right)$.

Each phase j uniformly explores classifiers learned based on samples from each distribution, collected up to phase $j-1$. Let $Q_i^{(j)}$ be the set of samples collected from D_i in phases $1, \ldots, j-1$. And let $h_i^{(j)} = \mathrm{ERM}_{\mathcal{H}}\left(Q_i^{(j)}\right)$. In each round of phase j, the agent picks one of $\{h_1^{(j)}, \ldots, h_k^{(j)}\}$ uniformly at random. Let h_j^* be the classifier of phase j with the highest sum of test accuracies. The agent can compute the test accuracy by dividing payment by the contract coefficient (which is non-zero in exploration phases).

The agent treats all rounds other than $\{i_1, \ldots, i_r\}$ as exploitation rounds. For an exploitation round t, suppose the last collected sample was in phase j. Then in round t, the agent selects the best classifier in phase $j-1$: h_{j-1}^*. Due to the completion of the $(j-1)^{\text{th}}$ exploration phase, h_{j-1}^* is guaranteed to have some optimality guarantees.

The regret analysis for this algorithm is presented in Appendix A.3.

□

Propositions 3 and 4 together imply that the principal can achieve $O(T^{3/4})$ $\mathcal{H}$-Regret through linear contracting with a rational agent. We next show that the principal cannot achieve better $\mathcal{H}$-regret rates when the agent's $\mathcal{H}$-regret rate is $O\left(\left(\sum_{t\in[T]} c_t\right)^{2/3}\right)$.

Proposition 5. *For any sequence of linear contracts $(c_t)_{t\in[T]}$, if the principal contracts with an agent with $O\left(\left(\sum_{t\in[T]} c_t\right)^{2/3}\right)$ $\mathcal{H}$-regret on all problem instances, the principal's $\mathcal{H}$-regret is $\Omega(T^{3/4})$ for some problem instance.*

Proof. Let us denote $\sum_{t=1}^{T} c_t$ by s_T. An agent with $O\left(s_T^x\right)$ $\mathcal{H}$-regret collects $O\left(s_T^x\right)$ samples since the agent incurs cost linear in the number of samples collected. This upper bound on the number of samples provides a lower bound on the excess errors of the deployed classifiers and therefore a lower bound on the principal's regret.

Usual sample complexity lower bounds provide lower bounds on the error of learning using a number of i.i.d samples. However, in our setting, the agent has more than just the samples he collects to learn classifiers. Through the linear payments he receives, he also has access to estimates of the accuracy of classifiers he provides in each round. This is a form of (noisy) query access to the distribution.

We provide a min-max lower bound on learning using both i.i.d samples and queries of the form answering the expected error on D^* of a classifier in the following proposition. We show that in a min-max sense, the queries do not allow for more accurate classifiers compared to using just the i.i.d samples.

Lemma 3 (Lower bound on error from using samples and queries)**.** *Consider a learning algorithm that uses m i.i.d samples and q queries of accuracies of classifiers. Then there exists a distribution D for which the expected error of the learned classifier is $\Omega(1/\sqrt{m})$ more than the optimal error of a one-dimensional halfspace on D.*

We prove this lemma in Appendix A.4. The main intuition for this lower bound is that without any structure on the distribution, it is hard to know what classifiers are useful to query. Therefore the queries are not useful.

Due to the above proposition, there is a problem instance for which the classifiers the principal deploys have an average excess error of $\Omega(1/s_T^{1/3})$ resulting in regret $\Omega(T/s_T^{1/3})$.

The principal also incurs a cost of order $\Omega(s_T)$ due to the contracting. Due to the sub-linearity of agent's regret in s_T, the payments have to be $\Omega(s_T)$ resulting in this term of the principal's regret.

The optimal choice of s_T to balance these two terms is $s_T \in \Theta(T^{3/4})$ resulting in regret of order $\Omega(T^{3/4})$.

□

Comparison with non-delegation benchmark. We have shown that the principal's $\mathcal{H}$-regret through delegating with an $O\left(\left(\sum_{t\in[T]} c_t\right)^{2/3}\right)$ $\mathcal{H}$-regret agent is $\Omega(T^{3/4})$. Since there is an algorithm for the agent to achieve $O\left(\left(\sum_{t\in[T]} c_t\right)^{2/3}\right)$ $\mathcal{H}$-regret, a rational agent is likely to achieve this regret upper bound.

If the principal instead performs this learning without delegation, the achievable regret is the same as the agent's achievable regret when the contract coefficient is 1 in every round. By Proposition 4, $O(T^{2/3})$ regret is achievable. Note that this means the $\mathcal{H}$-regret rate is strictly worse for the principal through delegation compared to without delegation.

6 Conclusions

Different parts of the learning pipeline are increasingly being delegated to autoML services and firms. Focusing on the delegation of data collection in such settings, we developed a principal-agent model capturing the practical challenges of hidden action and hidden state arising due to information asymmetry. We hope our work inspires future research exploring and addressing other incentive-theoretic obstacles that arise in this domain and in the delegation of other parts of the learning pipeline.

We identified the practicality of linear contracts under many problem parameters. We also obtained insights that hold for many other delegation settings, including the decreased utility and information rent the principal faces due to not knowing the hidden state. We also addressed a more realistic form of hidden state information asymmetry where the agent gradually learns the hidden state while executing the contract.

Many of our results rely on a specific structure for the dependence of error rates on number of samples used for training. These rates are exact for some learning tasks (see Remark 2) but are generally upper bounds. From the principal's perspective, the contracts we designed continue to have good accuracy guarantees even if the error rates are just upper bounds and not exact. The

agent's perspective is more complicated. Since the agent can learn more about the true error curve during learning and not rely on the upper bound, analyzing how the error curve learning occurs is needed to allow us to design truly incentive-compatible contracts in this more general setting. However, learning the error curve shape is learning from a much broader class and is likely to be more challenging. How this challenge impacts the utility of contracts would be an interesting direction for future work.

Acknowledgements

Funded in part by the European Union (ERC-2022-SYG-OCEAN-101071601). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council.

This work was supported in part by the National Science Foundation under grant CCF-2145898, by the Office of Naval Research under grant N00014-24-1-2159, a C3.AI Digital Transformation Institute grant, and Alfred P. Sloan fellowship, and a Schmidt Science AI2050 fellowship

References

Daron Acemoglu, Ali Makhdoumi, Azarakhsh Malekian, and Asu Ozdaglar. Too much data: Prices and inefficiencies in data markets. *American Economic Journal: Microeconomics*, 14(4):218–256, 2022.

Anish Agarwal, Munther Dahleh, and Tuhin Sarkar. A marketplace for data: An algorithmic solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 701–726, 2019.

Anish Agarwal, Munther Dahleh, Thibaut Horel, and Maryann Rui. Towards data auctions with externalities. *arXiv preprint arXiv:2003.08345*, 2020.

Tal Alon, Paul Dütting, and Inbal Talgam-Cohen. Contracts with private cost per unit-of-effort. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 52–69, 2021.

Tal Alon, Paul Dütting, Yingkai Li, and Inbal Talgam-Cohen. Bayesian analysis of linear contracts. *arXiv preprint arXiv:2211.06850*, 2022.

Stephen Bates, Michael I. Jordan, Michael Sklar, and Jake A. Soloff. Principal-agent hypothesis testing. *arXiv preprint arXiv:2205.06812*, 2022.

Curtis Bechtel, Shaddin Dughmi, and Neel Patel. Delegated Pandora's box. *arXiv preprint arXiv:2202.10382*, 2022.

Dirk Bergemann and Alessandro Bonatti. Markets for information: An introduction. *Annual Review of Economics*, 11:85–107, 2019.

Patrick Bolton and Mathias Dewatripont. *Contract Theory*. 2004.

Yang Cai, Constantinos Daskalakis, and Christos Papadimitriou. Optimum statistical estimation with strategic data sources. In *Conference on Learning Theory*, pages 280–296. PMLR, 2015.

Gabriel Carroll. Robustness and linear contracts. *American Economic Review*, 105(2):536–63, 2015.

Hector Chade and Jeroen Swinkels. Disentangling moral hazard and adverse selection. Technical report, Working Paper, Arizona State University.[450], 2019.

Junjie Chen, Minming Li, and Haifeng Xu. Selling data to a machine learner: Pricing via costly signaling. In *International Conference on Machine Learning*, pages 3336–3359. PMLR, 2022.

Lingjiao Chen, Paraschos Koutris, and Arun Kumar. Towards model-based pricing for machine learning in a data marketplace. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1535–1552, 2019.

Alessandro Chiesa and Tom Gur. Proofs of proximity for distribution testing. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

Paul Dütting, Tim Roughgarden, and Inbal Talgam-Cohen. Simple versus optimal contracts. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 369–387, 2019.

Paul Dütting, Tim Roughgarden, and Inbal-Talgam Cohen. The complexity of contracts. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms*, pages 2688–2707. SIAM, Philadelphia, PA, 2020.

Peter Frazier, David Kempe, Jon Kleinberg, and Robert Kleinberg. Incentivizing exploration. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, pages 5–22, 2014.

Shafi Goldwasser, Guy N Rothblum, Jonathan Shafer, and Amir Yehudayoff. Interactive proofs for verifying machine learning. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

Guru Guruganesh, Jon Schneider, and Joshua R Wang. Contracts under moral hazard and adverse selection. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 563–582, 2021.

Chien-Ju Ho, Aleksandrs Slivkins, and Jennifer Wortman Vaughan. Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. *Journal of Artificial Intelligence Research*, 55:317–359, 2016.

Hengrui Jia, Mohammad Yaghini, Christopher A Choquette-Choo, Natalie Dullerud, Anvith Thudi, Varun Chandrasekaran, and Nicolas Papernot. Proof-of-learning: Definitions and practice. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1039–1056. IEEE, 2021.

Jon Kleinberg and Robert Kleinberg. Delegated search approximates efficient search. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 287–302, 2018.

Jean-Jacques Laffont and David Martimort. *The Theory of Incentives*. Princeton University Press, 2009.

Zhiyuan Liu, Huazheng Wang, Fan Shen, Kai Liu, and Lijun Chen. Incentivized exploration for multi-armed bandits under reward drift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4981–4988, 2020.

Esther Rolf, Theodora T Worledge, Benjamin Recht, and Michael Jordan. Representation matters: Assessing the importance of subgroup allocations in training data. In *International Conference on Machine Learning*, pages 9040–9051. PMLR, 2021.

Eden Saig, Inbal Talgam-Cohen, and Nir Rosenfeld. Delegated classification. *arXiv preprint arXiv:2306.11475*, 2023.

Chris Ying, Aaron Klein, Eric Christiansen, Esteban Real, Kevin Murphy, and Frank Hutter. Nas-bench-101: Towards reproducible neural architecture search. In *International Conference on Machine Learning*, pages 7105–7114. PMLR, 2019.

A Omitted proofs

A.1 Proof of Proposition 2

Proof. The linear contract c^* that achieves this approximately optimal utility is the following:

$$c^* = \max\left(\frac{1}{\beta(p+1)^{\frac{p+1}{p}}}, \frac{\alpha d^{\frac{1}{p}}}{p} \cdot \left(\frac{p+1}{1-\theta}\right)^{\frac{p+1}{p}}\right).$$

We first show that this contract satisfies the participation constraint for the agent. When the linear contract is c times the accuracy, the agent the number of samples n to maximize the agent's utility $c\left(1-\theta-\frac{d}{n^p}\right)-\alpha n$. The number of samples the agent chooses as a function of c is $\left(\frac{cdp}{\alpha}\right)^{\frac{1}{p+1}}$. The contract c satisfies the participation constraint if the utility from choosing this number of samples is non-negative. This utility is:

$$\begin{aligned}
&= c\left(1-\theta-d\left(\frac{\alpha}{cdp}\right)^{\frac{p}{p+1}}\right) - \alpha\left(\frac{cdp}{\alpha}\right)^{\frac{1}{p+1}} \\
&= c(1-\theta) - c^{\frac{1}{p+1}}\left(\frac{\alpha d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}} - c^{\frac{1}{p+1}}\left(\alpha d^{\frac{1}{p}}{}^{\frac{p}{p+1}}\right) p^{\frac{1}{p+1}} \\
&= c(1-\theta) - c^{\frac{1}{p+1}} \cdot \frac{p+1}{1-\theta} \cdot \left(\frac{\alpha d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}}.
\end{aligned}$$

This utility is non-negative when $c \geq \frac{\alpha d^{\frac{1}{p}}}{p}\left(\frac{p+1}{1-\theta}\right)^{\frac{p+1}{p}}$. By the definition of c^*, it is greater than $\frac{\alpha d^{\frac{1}{p}}}{p}$ and so c^* satisfies the participation constraint.

When the principal chooses a linear contract c, it achieves a utility

$$U_{\text{lin}}(c) = (1-\beta c)\left(1-\theta-\left(\frac{\alpha d^{\frac{1}{p}}}{pc}\right)^{\frac{p}{p+1}}\right).$$

We can provide an upper bound on the optimum utility using the optimum utility of the principal when there is no noise in the observed accuracy. In this case, the principal gets the agent to collect $\left(\frac{dp}{\alpha\beta}\right)^{\frac{1}{p+1}}$ and pays the agent α times this amount. So the optimum utility U_{opt} satisfies

$$U_{\text{opt}} \leq 1-\theta-(p+1)\left(\frac{\alpha\beta d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}}.$$

To show that c^* achieves the approximation guaranteed in the theorem, we consider two cases.

Case 1. The first case is when $c^* = \frac{1}{\beta(p+1)^{\frac{p+1}{p}}}$. In this case the utility of c^* is

$$\begin{aligned}
U_{\text{lin}}(c^*) &= \left(1-\frac{1}{(p+1)^{\frac{p+1}{p}}}\right)\left(1-\theta-(p+1)\left(\frac{\alpha\beta d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}}\right) \\
&\geq \left(1-\frac{1}{(p+1)^{\frac{p+1}{p}}}\right) U_{\mathbf{opt}}
\end{aligned}$$

So in the first case, we have shown the required bound on the approximation ratio of the contract c^*.

Case 2. The other case is when $c^* = \frac{\alpha d^{\frac{1}{p}}}{p}\left(\frac{p+1}{1-\theta}\right)^{\frac{p+1}{p}}$. In this case,

$$U_{\text{lin}}(c^*) = \left(1 - \frac{\alpha\beta d^{\frac{1}{p}}}{p}\left(\frac{p+1}{1-\theta}\right)^{\frac{p+1}{p}}\right)\cdot\frac{(1-\theta)p}{p+1}$$

$$\frac{U_{\text{lin}}(c^*)}{U_{\text{opt}}} = \frac{\left(1 - \frac{\alpha\beta d^{\frac{1}{p}}}{p}\left(\frac{p+1}{1-\theta}\right)^{\frac{p+1}{p}}\right)\cdot\frac{(1-\theta)p}{p+1}}{(1-\theta)\left(1 - \frac{p+1}{1-\theta}\left(\frac{\alpha\beta d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}}\right)}$$

$$= \frac{\frac{p}{p+1}\left(1 - t^{\frac{p+1}{p}}\right)}{1-t} \qquad \left(\text{where } t = \frac{p+1}{1-\theta}\cdot\left(\frac{\alpha\beta d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}}\right).$$

$U_{\text{lin}}(c^*)/U_{\text{opt}}$ is increasing in t. The condition of this case 2 occurring turns out to be a condition on t. This condition for case 2 occurring is the following:

$$\frac{1}{\beta(p+1)^{\frac{p+1}{p}}} \le \frac{\alpha d^{\frac{1}{p}}}{p}\left(\frac{p+1}{1-\theta}\right)^{\frac{p+1}{p}}$$

$$\implies \frac{1}{p+1} \le \frac{p+1}{1-\theta}\cdot\left(\frac{\alpha\beta d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}}$$

$$\implies \frac{1}{p+1} \le t.$$

Case 2 occurs when $t \ge 1/(p+1)$ and in this case $U_{\text{lin}}(c^*)/U_{\text{opt}}$ is increasing in t. So the smallest value of $U_{\text{lin}}(c^*)/U_{\text{opt}}$ in this case occurs when $t = 1/(p+1)$. Plugging in this value, we get that for this case,

$$\frac{U_{\text{lin}}(c^*)}{U_{\text{opt}}} \ge 1 - \frac{1}{(p+1)^{\frac{p+1}{p}}}$$

□

A.2 Proof of Theorem 2

Proof. Recall that the first-best contract has a threshold form. The contract offers payment t^* when the test error is less than or equal to ℓ^* and offers payment zero otherwise. Let us denote the sample complexity to get expected loss at most ℓ by $n(\ell)$. That is,

$$n(\ell) = \left(\frac{d}{\ell - \theta}\right)^{1/p}.$$

The optimal contract offers $t^* = \alpha n(\ell^*)$ where α is the cost per sample for the agent. Let us denote $n(\ell^*)$ by n^*. And $n^* = (pd/\alpha\beta)^{1/(p+1)}$.

We will show that the best response for the agent against this contract is never to collect samples less than $n(\ell+\epsilon)$ when the test set has size $O\left(\frac{1}{\epsilon^2}\log\frac{d}{\epsilon}\right)$. We show this by showing that the agent's utility in choosing $n(\ell+\Delta)$ is less than the agent's utility in collecting $n(\ell-\Delta)$ for all $\Delta > \epsilon$.

The number of samples the agent would collect to get expected error $l^* + \Delta$ is such that:

$$\theta + \frac{d}{n_1^p} = \theta + \frac{d}{n^{*p}} + \Delta$$
$$n_1 = \frac{n^* d^p}{(d+\Delta)^p}.$$

Similarly, the number of samples needed to get expected error $l^* - \Delta$ is

$$n_2 = \frac{n^* d^p}{(d-\Delta)^p}.$$

For any action of the agent, the probability that the observed loss is ϵ or more away from the expected loss is less than $2\exp(-2m\epsilon^2)$. This is by applying Hoeffding's inequality on the observed loss random variable which is bounded between 0 and 1. As a result, for $\Delta > \epsilon$, the expected payment when collecting n_1 and n_2 samples is $\leq 2t^* \exp(-2m\epsilon^2)$ and $\geq t^*(1 - 2\exp(-2m\epsilon^2))$ respectively. The agent's utility due to n_1 is less than the utility due to n_2 when

$$\alpha n^* \left(1 - 4\exp(-2m\epsilon^2)\right) \geq \alpha n^* d^p \left(\frac{1}{(d - 2\Delta n^{*1/p})^p} - \frac{1}{(d + 2\Delta n^{*1/p})^p}\right).$$

Let us denote $\kappa = d^p \left(\frac{1}{(d-2\Delta n^{*1/p})^p} - \frac{1}{(d+2\Delta n^{*1/p})^p}\right)$. So this occurs when

$$m \geq \frac{1}{2\epsilon^2} \log \frac{4}{1-\kappa}.$$

Note that $\frac{1}{1-\kappa}$ is polynomial in both d and $\frac{1}{\epsilon}$.

□

A.3 Proof of Proposition 4

Proof. We start by restating the agent's algorithm. For any round t, let us denote $\sum_{t' \leq t} c_t$ by s_t. Since the agent incurs cost linear in the number of samples collected, the algorithm must limit the number of samples to $O\left(s_T^{2/3}\right)$. To do this, at round t, the agents collects $\max\left(0, \left\lfloor s_t^{2/3} \right\rfloor - \left\lfloor s_{t-1}^{2/3} \right\rfloor\right)$ samples. As a result, for any t, the number of samples collected up to round t is $\lfloor s_t^{2/3} \rfloor$.To select the distribution to sample from, the agent picks one out of $\{D_1, \ldots, D_k\}$ uniformly at random. This fully describes how the agent's algorithm collects samples by describing the number of samples and distributions to sample from.

The remaining component of the agent's algorithm is classifier selection. The classifier selection includes exploration and exploitation where the exploration is to determine which of $D_1, \ldots, D_k$ provides samples to train an accurate classifier. Since we know D^* is one of these k distributions, we are guaranteed that there exists a distribution providing relevant samples.

We assign rounds of sample collection to be rounds of exploration. These rounds are suitable for exploration for two reasons. The first is because the principal deploys the agent-provided classifiers in these rounds. By the algorithm's design, the agent only collects samples in rounds with $c_t > 0$, and in feedback-providing contracts, agent-provided classifiers are deployed in these rounds. Secondly, this choice also leads to the right number of exploration rounds for the optimal exploration-exploitation tradeoff. Let $\{i_1, \ldots, i_r\}$ denote the indices of rounds of exploration. This set's size is at most $\left(\sum_{t \in [T]} c_t\right)^{2/3}$. A phased exploration is done in these rounds.

The phased exploration divides the indices $\{i_1, \ldots, i_r\}$ into phases where phase j is of length 2^{j-1}. So at least 2^{j-1} samples are collected in phase j. The number of phases is therefore $\log\left(\left(\sum_{t\in[T]} c_t\right)^{2/3}\right)$.

Each phase j uniformly explores classifiers learned based on samples from each distribution, collected up to phase $j-1$. Let $Q_i^{(j)}$ be the set of samples collected from D_i in phases $1, \ldots, j-1$. And let $h_i^{(j)} = \text{ERM}_{\mathcal{H}}\left(Q_i^{(j)}\right)$. In each round of phase j, the agent picks one of $\{h_1^{(j)}, \ldots, h_k^{(j)}\}$ uniformly at random. Let h_j^* be the classifier of phase j with the highest sum of test accuracies. The agent can compute the test accuracy by dividing payment by the contract coefficient (which is non-zero in exploration phases).

The agent treats all rounds other than $\{i_1, \ldots, i_r\}$ as exploitation rounds. For an exploitation round t, suppose the last collected sample was in phase j. Then in round t, the agent selects the best classifier in phase $j-1$: h_{j-1}^*. Due to the completion of the $(j-1)^{\text{th}}$ exploration phase, h_{j-1}^* is guaranteed to have some optimality guarantees.

Regret analysis. Now we analyze the regret incurred due to this algorithm. The regret due to cost of sampling is $O(s_T^{2/3})$. We will analyze the regret due to the sub-optimality of the classifiers deployed. Let $\theta^* = \min_{h\in\mathcal{H}} L_{D^*}(h)$. Since the number of exploration rounds is $O(s_T^{2/3})$, the regret in these rounds is trivially bounded by $O(s_T^{2/3})$. We will focus on sub-optimality of classifiers deployed in the exploitation rounds.

Consider the classifier h_j^* yielding the highest accuracy in phase j. Let $L_j(h)$ indicate the average test loss of deploying classifier h in phase j. Each of the candidate classifiers of phase j: $\{h_1^{(j)}, \ldots, h_k^{(j)}\}$ is explored $2^{j-1}/k$ times since we perform uniform exploration. Therefore $\left|L_{D^*}(h_i^{(j)}) - \mathbb{E}[L_j(h_i^{(j)})]\right| \leq \sqrt{k/2^{j-1}}$ and $L_{D^*}(h_j^*) \leq \theta^* + 2\sqrt{k/2^{j-1}}$.

Let t_j be the last non-sampling round that h_{j-1}^* is selected. This means that t_j+1 is a sampling round and of phase $j+1$. The number of samples drawn up to phase j and hence up to round t_j is $\geq 2^{j+1}$. Due to this, t_j+1 being a sampling round implies that $\sum_{i<t_i} c_i \leq 2^{j+1}$. So we can bound the regret due to deploying h_{j-1}^* in non-sampling rounds to be at most

$$\sum_{i\leq t_j} c_i(L_{D^*}(h_{j-1}^*) - \theta^*) \leq \frac{2^{j+1}\sqrt{k}}{2^j}$$
$$\in O(\sqrt{2^j})$$

Summing over regret due to non-sampling rounds over all phases j from 1 to $\log(s_T^{2/3})$, total regret in non-sampling rounds is at most: $\sum_{j=1}^{\log(s_T^{2/3})} O(\sqrt{2^j}) \in O(s_T^{2/3})$.

Therefore the total regret over sampling and non-sampling rounds is $O(s_T^{2/3})$.

□

A.4 Proof of Lemma 3

Proof. For each n, q, we will construct a class of distributions such that for any learning algorithm with access to n i.i.d. samples and q queries, there is a distribution D^* in the class for which the learning algorithm will have excess error at least $\theta^* + \Omega(1/\sqrt{n})$ where θ^* is the optimal error achieved by the class of one-dimensional half-spaces on D^*. This is the class $\mathcal{H}_{1d-HS}$ which we will refer by $\mathcal{H}$

$$\mathcal{H}_{1d-HS} = \{\mathbb{1}\{x \geq \theta\} : \theta \in \mathbb{R}\} \cup \{\mathbb{1}\{x \leq \theta\} : \theta \in \mathbb{R}\}.$$

As a construction for the lower bound, consider the class of distributions over a domain of M points, each having a uniform marginal distribution supported on $m < M$ of those points. The labelling distribution $\Pr(y = 1|x)$ is $1/2 \pm 1/2\sqrt{n}$. We later describe how to choose m, M so that the lower bound holds for n, q.

We will show that for any set of q queries, there are two distributions D_1, D_2 such that

1. All query values are the same for D_1, D_2.
2. No algorithm can distinguish between D_1, D_2 with probability more than $1/2$ using n samples drawn.
3. Any classifier with error $\min_{h \in \mathcal{H}_{1d-DS}} L_{D_1}(h) + O(1/\sqrt{n})$ on D_1 necessarily has error $\min_{h \in \mathcal{H}_{1d-DS}} L_{D_2}(h) + \Omega(1/\sqrt{n})$ on D_2.

The above properties suffice to show the required lower bound. This is because with the above properties, a learning algorithm that achieves $o(1\sqrt{n})$ expected excess error necessarily distinguishes between D_1 and D_2 with probability at least $1/2$. Distinguishing between D_1, D_2 should not be possible with n samples and q queries if the above properties hold.

Now let us show how to choose m, M and construct D_1, D_2 to make the above properties hold.

Each query assigns a value of 0 or 1 to each point. So there is a sequence of length q indicating the labels the queries assign for each point. We can partition the domain into points having the same sequence of query labels. By constructing D_1, D_2, so that the number of points in each partition with labelling function $1/2 + 1/2\sqrt{n}$ is the same for D_1 and D_2, we can guarantee that all query values are the same for D_1 and D_2

Let us set M so that $M > 2^q m$. Since there are 2^q query sequences, at least one of the partition sets has size m. Consider the m points distributions D_1, D_2 are supported on to be the n points of samples drawn and the remaining points are points from the partition of size m. There are at least $m - n$ points of the support from the partition of size m.

Let the points in the support that are in the query partition of size m be $x_1 \leq \ldots \leq x_s$ where $s \geq m - n$. D_1 has probability of $+1$ label $1/2 + 1/\sqrt{n}$ for points $x_1, \ldots, x_{s/2}$ and probability of $+1$ label $1/2 - 1/\sqrt{n}$ for points $x_{s/2+1}, \ldots, x_m$. D_2 has probability of $+1$ label $1/2 - o1/\sqrt{n}$ for points $x_1, \ldots, x_{s/2}$ and probability of $+1$ label $1/2 + 1/\sqrt{n}$ for points $x_{s/2+1}, \ldots, x_m$.

Outside of this partition, let the labelling distribution of any other point in the support be the same for D_1, D_2. By this construction, property (1) is satisfied.

Restricted to the query partition, the errors on distributions D_1, D_2 sum up to $1/2 + 1/\sqrt{n}$. By choosing $m > 2n$, the query partition makes up at least half the fraction of the support. Therefore any classifier h has $L_{D_1}(h) + L_{D_2}(h) = 1/4 + 1/2\sqrt{n}$. and due to this property (3) holds.

The KL divergence between the sampling distributions from D_1, D_2 is at most $\sqrt{n}\frac{1}{2\sqrt{n}}$, and therefore with probability $\geq 1/2$, we cannot distinguish between D_1 and D_2 using n samples. This shows property (2) holds. □

B Miscellaneous results and discussions

B.1 Treating error curves as upper bounds

For most of our results we have made use of the structured form of error curves reflecting how expected error of a learned model is assumed to vary with the number of samples used for training. This structure is inspired by statistical minimax bounds and are upper bounds rather the true error

curves. We designed contracts assuming the bounds to be actual error curves. Here we discuss what we can say about these contracts without assuming the bounds to be exact error curves.

From the principal's perspective, these contracts result in accuracy that is just as good as that of learned models. However, the principal would end up paying the agent more than it could have if the principal knew the exact error curve. We can view the shape of the true error curve as another piece of information the principal is unaware of in addition to the optimal error. This hidden information results in more information rent but does not impact the accuracy of the model obtained from delegation.

The agent's perspective of what changes is more complicated. Our contracts assumed that the agent responded assuming that the upper bound was the true error curve. It may be reasonable that before starting the delegation process, the agent believes the upper bounds to be the true curves having no other frame of reference. However after starting to collect data, it is possible that the agent will learn more about the form of the true error curve and respond differently. This is similar to how the agent can learn the optimal error while executing the contract. Analyzing how this error curve learning occurs will allow us to design truly incentive-compatible contracts. However, learning the error curve shape is learning from a much broader class and is likely to be more challenging.

B.2 Variable label quality model

The setting above models the scenario where the agent does not have the option to choose the quality of the data is collects. However, the agent might be able to control the quality of the data as a function of the cost per sample. We study a model of quality of data where the quality corresponds to the quality of the labels of the data. The quality parameter $q = 1 - 2\eta$ captures the likelihood of the labels being correct. Here, $\eta \in (0, 1/2)$ is the probability of the label being incorrect. In this setting, the expected accuracy on the principal's test set from the agent collecting n samples at quality level q when the optimal error is θ is $1 - \theta - \frac{1}{qn^p}$ for some $p > 0$. We assume that the cost of collecting a single sample at quality level q for the agent is given by $\alpha(q)$ a function increasing in q and convex. So the cost for the agent of collecting n samples at quality level q is $C(n, q) = \alpha(q)n$.

In this section, we provide results for $\alpha(q) = q^b + \alpha_0$ for $b > 0$. We can think of α as the cost of collecting an unlabelled sample and q^b as the cost of labelling an unlabelled point. The main message of this section is that even though the quality of labels is an action that the agent chooses, effectively, this choice is not information that is private from the principal. It turns out that whatever the contract is, the utility-maximizing agent executes the contract by choosing a single quality value q^*. The principal can also compute q^* so the quality is not a reflection of information asymmetry. Therefore, this regime is essentially the same as the one studied in the previous section.

Theorem 4 (Constant quality level)**.** *For any problem parameters $d, p, \alpha_0 > 0, b > 1$, when $\alpha(q) = q^b + \alpha_0$, for any expected accuracy a the agent wishes to achieve, the agent chooses a constant q^* that only depends on α_0, b as the quality level.*

Proof. If the agent aims to achieve an expected accuracy of at least s, then the agent chooses the number of samples and quality level by solving the following optimization problem:

$$\begin{aligned} \min_{q,n} \quad & \left(q^b + \alpha_0\right) n \\ \text{s.t.} \quad & \theta + \frac{1}{qn^p} \leq 1 - a \\ & 0 \leq q \leq 1. \end{aligned}$$

The solution of this optimization problem can be calculated as follows:

$$\begin{aligned}\mathcal{L} &= \left(q^b + \alpha\right) n + \lambda_1 \left(\theta - l + \frac{1}{qn^p}\right) + (\lambda_2 - \lambda_3)q \\ \nabla_n \mathcal{L} &= q^b + \alpha - \frac{p\lambda_1}{bn^{p+1}} \\ \nabla_q \mathcal{L} &= bq^{b-1}n - \frac{\lambda_1}{q^2 n^p} + \lambda_2 - \lambda_3.\end{aligned}$$

If $\lambda_2^* = \lambda_3^* = 0$, we obtain

$$\begin{aligned}&\implies \lambda_1^* = bq^{*b+1}n^{p+1} \\ &\implies q^* = \left(\frac{\alpha}{b-1}\right)^{\frac{1}{b}},\end{aligned}$$

and n^* is obtained by solving

$$\theta + \frac{1}{q^* n^{*p}} = l.$$

If $(\alpha/(b-1))^{1/b}$ is not in $[0,1]$, then λ_2^* or λ_3^* is non-zero and q^* is either 0 or 1. □

B.3 Closed-form solution for the two optimal error, hidden state problem

Here we solve the state-aware optimization problem Opt when there are two optimal errors, $\underline{\theta} \leq \overline{\theta}$, with a prior probability of ν for $\underline{\theta}$. Let $L(n) = d/n^p$. We solve the following optimization problem and show that the solution is the optimal solution we are looking for. Note that this problem omits the incentive-compatibility constraint for the problem $\overline{\theta}$ and the participation constraint for the easy problem.

$$\begin{aligned}\min_{\underline{n},\overline{n},\underline{t},\overline{t}} \quad & \nu(L(\underline{n}) + \beta\underline{t}) + (1-\nu)(L(\overline{n}) + \beta\overline{t}) \\ \text{s.t.} \quad & \overline{t} - \frac{\alpha\overline{n}}{(1+\Delta\overline{n}^p)^{1/p}} - \underline{t} + \alpha\underline{n} \leq 0 \\ & \alpha\overline{n} - \overline{t} \leq 0.\end{aligned}$$

First note that this is a convex optimization problem, where the objective is convex by the convexity of the loss and the PC constraint is a linear constraint. All that is left is to check that the IC constraint is convex. This is the sum of linear terms and the term $\frac{-\alpha d^{1/p}\overline{n}}{(d+\Delta\overline{n}^p)^2}$. The second derivative of this term is $\frac{3\alpha d^{1/p}\Delta}{2\overline{n}^p(d+\Delta\overline{n}^p)^4}$. Since the second derivative is positive, the IC constraint is convex.

Consider the Lagrangian

$$L(\underline{n},\overline{n},\underline{t},\overline{t};\lambda_1,\lambda_2) = \nu(L(\underline{n})+\beta\underline{t})+(1-\nu)(L(\overline{n})+\beta\overline{t})+\lambda_1\left(\overline{t} - \frac{\alpha d^{1/p}\overline{n}}{(d+\Delta\overline{n}^p)^{1/p}} - \underline{t} + \alpha\underline{n}\right)+\lambda_2\left(\alpha\overline{n} - \overline{t}\right),$$

which has the following gradients:

$$\nabla_{\underline{n}} L = \nu L'(\underline{n}) + \alpha\lambda_1 \tag{G1}$$
$$\nabla_{\underline{t}} L = \nu\beta - \lambda_1 \tag{G2}$$
$$\nabla_{\overline{t}} L = (1-\nu)\beta - \lambda_2 + \lambda_1 \tag{G3}$$
$$\nabla_{\overline{n}} L = (1-\nu)L'(\overline{n}) + \alpha\lambda_2 - \lambda_1 \left(\frac{\alpha d^{1/p}}{(d + \Delta\overline{n}^p)^{(p+1)/p}} \right) \tag{G4}$$

To choose values $\underline{n}^*, \overline{n}^*, \underline{t}^*, \overline{t}^*, \lambda_1^*, \lambda_2^*$ that satisfy the KKT conditions, first we set the gradients to zero:

$$L'(\underline{n}^*) = -\alpha\beta \qquad \text{(From (G1), (G2))}$$
$$\lambda_1^* = \nu\beta \qquad \text{(From (G2)}$$
$$\lambda_2^* = \beta \qquad \text{(From (G3) and value of } \lambda_1^*)$$
$$(1-\nu)L'(\overline{n}^*) + \alpha\beta - \frac{\nu\alpha\beta}{(1+\Delta\overline{n}^p)^{(p+1)/p}} = 0 \qquad \text{(From (G4) and values of } \lambda_1^*, \lambda_2^*)$$
$$\implies L'(\overline{n}^*) = -\frac{\alpha\beta}{1-\nu}\left(1 - \frac{\nu d^{1/p}}{(d+\Delta\overline{n}^{*p})^{(p+1)/p}}\right).$$

By complementary slackness,

$$\alpha\overline{n}^* = \overline{t}^*$$
$$\underline{t}^* = \alpha\left(\underline{n}^* - \frac{d^{1/p}\overline{n}^*}{(d+\overline{n}^{*p})^{1/p}} + \overline{n}^*\right).$$

The contract described by $(\underline{n}^*, \overline{n}^*, \underline{t}^*, \overline{t}^*)$ satisfies the properties of the second-best contract in the classical contract theory setting. We list these properties here:

P1 No output distortion for the easy problem: $\underline{n}^*$ is the solution of $L'(\underline{n}^*) = -\alpha\beta$ which is also the value of n^{fb}. So for the easy problem, the agent gathers the same number of samples as in the full information case.

P2 Downward distortion for the hard problem:

$$\begin{aligned} L'(\overline{n}^*) &= -\frac{\alpha\beta}{1-\nu}\left(1 - \frac{\nu d^{1/p}}{(d+\Delta\overline{n}^{*p})^{(p+1)/p}}\right) \\ &< -\alpha\beta \\ &= L'(n^{fb}). \end{aligned}$$

So $\overline{n}^* < n^{fb}$. For the hard problem, the agent gathers fewer samples than in the full information case.

P3 When the problem is easy, the agent gets positive information rent:

$$\begin{aligned} \underline{t}^* - \alpha\underline{n}^* &= \overline{n}^* - \frac{d^{1/p}\overline{n}^*}{(d+\Delta\overline{n}^{*p})^{(p+1)/p}} \\ &> 0. \end{aligned}$$

We now check that the contract that is the solution to the above optimization problem also satisfies the omitted constraints. First we start with the participation constraint for the easy problem. By the positive information rent property (P3) we know that $\alpha \underline{n}^* < \underline{t}^*$. Next consider the incentive-compatibility constraint for the hard problem. We only need to check when $\Delta \underline{n}^{*p} < d$. Otherwise, the IC constraint automatically holds. The difference in agent's utility between choosing the $\overline{\ell}^* = \overline{\theta} + d/\overline{n}^{*p}$ option and the $\underline{\ell}^* = \underline{\theta} + d/\underline{n}^{*p}$ option is:

$$
\begin{aligned}
&- \alpha \underline{n}^* + \bar{t} + \frac{\alpha d^{1/p} \underline{n}^*}{(d - \Delta \underline{n}^{*p})^p} - \underline{t}^* \\
&= -\frac{\alpha d^{1/p} \overline{n}^*}{(d + \Delta \overline{n}^{*p})^{1/p}} + \alpha \overline{n}^* + \frac{\alpha d^{1/p} \underline{n}^*}{(d - \Delta \underline{n}^{*p})^{1/p}} - \alpha \underline{n}^* && \text{(Using values of } \underline{t}^*, \bar{t}^*) \\
&= \alpha \left(\frac{d^{1/p} \underline{n}^*}{(d - \Delta \underline{n}^{*p})^{1/p}} - \underline{n}^* + \frac{d^{1/p} \overline{n}^*}{(d + \Delta \overline{n}^{*p})^{1/p}} - \overline{n}^* \right) \\
&\geq \alpha \overline{n}^* \left(\frac{1}{(1 - \Delta \underline{n}^{*p})^{1/p}} + \frac{1}{(1 + \Delta \underline{n}^{*p})^{1/p}} - 2 \right) && \text{(Since } \underline{n}^* < \overline{n}^*)
\end{aligned}
$$

Note that the function $\frac{d^{1/p}}{(d-x)^q} + \frac{d^{1/p}}{(d+x)^q}$ is increasing in the interval $[0, 1)$ for every q. The derivative of that function is $q \left(\frac{d^{1/p}}{(d-x)^{q+1}} - \frac{d^{1/p}}{(d+x)^{q+1}} \right)$. This is nonnegative and lies in $[0, 1)$.

$$
\geq 0 \qquad \text{(Since we assume } 0 < \Delta \underline{n}^{p*} < 1).
$$

This solution finds the optimal contract under hidden state.

B.3.1 Separating contracts

To be able to compute any separating contract, it suffices to solve the above optimization problem with the additional constraint $\underline{n}, \overline{n} \geq n_0$ for some $n_0 \geq 0$. The new optimizers $\underline{n}(n_0), \overline{n}(n_0), \underline{t}(n_0).\bar{t}(n_0)$ are as follows:

$$
\begin{aligned}
\underline{n}(n_0) &= \max(\underline{n}^*, n_0) \\
\overline{n}(n_0) &= \max(\overline{n}^*, n_0) \\
\underline{t}(n_0) &= \alpha \underline{n}(n_0) \\
\bar{t}(n_0) &= \alpha \left(\underline{n}(n_0) - \frac{d^{1/p} \overline{n}(n_0)}{(d + \overline{n}(n_0))^{1/p}} + \overline{n}(n_0) \right).
\end{aligned}
$$

B.3.2 Optimal pooling contract

The optimal pooling contract optimizes over $\overline{n}$. $t = \alpha \overline{n}$. $\underline{n}$ is chosen such that

$$
\begin{aligned}
\underline{\theta} + \frac{d}{\underline{n}^p} &= \overline{\theta} + \frac{d}{\overline{n}^p} \\
\implies \underline{n} &= \frac{d^{1/p} \overline{n}}{(d + \Delta \overline{n}^p)^{1/p}}.
\end{aligned}
$$

Thus the optimization problem is choosing $\overline{n}$ to be the minima of

$$
\nu \frac{d}{\left(\frac{d^{1/p} n}{(d + \Delta n^p)^{1/p}} \right)^p} + (1 - \nu) \frac{d}{n^p} + \alpha \beta n.
$$

B.4 Tightness of linear contracts approximation

Our main result (Theorem 1) gave a linear contract that provably approximates the optimal contract up to a constant factor. This approximation factor stated in Proposition 2 is also tight as stated in the following theorem, which shows that there is a problem instance for which no linear contract can do better than a given approximation factor. The problem instance for which the approximation ratio is tight is one that has deterministic test error distribution, which arises when the size of the test set tends to infinity.

Theorem 5 (Tightness of approximation bound)**.** *For every $\theta \in [0, 1), p, d > 0$, there are problem parameters $\alpha, \beta > 0$ such that for the problem instance with these parameters, all linear contracts have at most $1 - \frac{1}{(p+1)^{\frac{p+1}{p}}}$ times the optimal utility.*

Proof. We will show that there exist α, β such that the contract $\frac{1}{\beta(p+1)^{\frac{p+1}{p}}}$ is the optimal contract chosen by the principal. This contract will also satisfy the participation constraint for our chosen values of α, β. Recall that the participation constraint is

$$\begin{aligned}\frac{1}{\beta(p+1)^{\frac{p+1}{p}}} &\geq \frac{\alpha d^{\frac{1}{p}}}{p} \cdot \left(\frac{p+1}{1-\theta}\right)^{\frac{p+1}{p}} \\ &\equiv 1 - \theta \geq \left(\frac{\alpha\beta d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}} (p+1)^2.\end{aligned}$$

The principal chooses the contract that sets the derivative of the above quantity to zero as long as that contract satisfies the participation constraint. If setting $\frac{1}{\beta(p+1)^{\frac{p+1}{p}}}$ yields a zero derivative and it satisfies the participation constraint, then it is the optimal linear contract. The derivative relative to c is

$$(1 - \beta c)\left(\frac{\alpha d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}} \cdot \frac{p}{p+1} \cdot \frac{1}{c^{\frac{2p+1}{p+1}}} - \beta\left(1 - \theta - \left(\frac{\alpha d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}}\right).$$

Setting $c = \frac{1}{\beta(p+1)^{\frac{p+1}{p}}}$, the derivative is

$$\begin{aligned}&\left(1 - \frac{1}{(p+1)^{\frac{p+1}{p}}}\right)\frac{p}{p+1}\left(\frac{\alpha d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}} \beta^{\frac{2p+1}{p+1}}(p+1)^{\frac{2p+1}{p}} \\ &- \beta\left(1 - \theta - (p+1)\left(\frac{\alpha\beta d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}}\right).\end{aligned}$$

We can choose α, β to set this derivative to zero by choosing α, β satisfying:

$$1 - \theta = \left(\frac{\alpha\beta d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}} (p+1)\left(1 + \left(1 - \frac{1}{(p+1)^{\frac{p+1}{p}}}\right) p(p+1)^{\frac{1}{p}}\right).$$

Note that for every $p > 0$, $\left(1 + \left(1 - \frac{1}{(p+1)^{\frac{p+1}{p}}}\right) p(p+1)^{\frac{1}{p}}\right) > p + 1$. So,

$$1 - \theta \geq \left(\frac{\alpha\beta d^{\frac{1}{p}}}{p}\right)^{\frac{p}{p+1}} (p+1)^2.$$

This shows that there are problem parameters that make $c^* = \frac{1}{\beta(p+1)^{\frac{p+1}{p}}}$ the optimal linear contract. In the proof of Proposition 2, we showed that this linear contract achieves at least $1 - \frac{1}{(p+1)^{\frac{p+1}{p}}}$ times the optimum utility. When the problem involves a deterministic mapping between the number of samples and the observed accuracy, this ratio is exact. □