Selling Data To a Machine Learner: Pricing via Costly Signaling

Junjie Chen [*1] **Minming Li** [*1] **Haifeng Xu** [*2]

Abstract

We consider a new problem of selling data to a machine learner who looks to purchase data to train his machine learning model. A key challenge in this setup is that neither the seller nor the machine learner knows the true *quality of data*. When designing a revenue-maximizing mechanism, a data seller faces the tradeoff between the *cost* and *precision* of data quality estimation. To address this challenge, we study a natural class of mechanisms that price data via costly signaling. Motivated by the assumption of i.i.d. data points as in classic machine learning models, we first consider selling homogeneous data and derive an optimal selling mechanism. We then turn to the sale of heterogeneous data, motivated by the sale of multiple data sets, and show that 1) on the negative side, it is NP-hard to approximate the optimal mechanism within a constant ratio $\frac{e}{e+1}+o(1)$; while 2) on the positive side, there is a $\frac{1}{k}$-approximate algorithm, where k is the number of the machine learner's private types.

1. Introduction

Last decade has witnessed rapid advances made in machine learning (ML), especially the deep learning field (Pouyanfar et al., 2018). The key factor driving these breakthroughs is the explosive growth of data over the Internet. Nowadays, from startups to giant companies like Microsoft and Alibaba, industries deploy trained ML models to their business operations and realize that relevant training data of good quality play a vital role in obtaining good performance of these models. As data now are becoming increasingly valuable, a new type of economy, i.e., data economy, is receiving more attention (Mehta et al., 2019; Bergemann et al., 2019).

Given the importance of data for machine learning nowadays, we take the first step to study a new data selling scenario: a seller (she) sells data to a machine learner (he), i.e., a buyer who would like to purchase data to train his ML model. It is worth pointing out that selling data is a problem different from selling items in the traditional auction model, in which items are usually indivisible. As pointed out by previous works (Agarwal et al., 2019; Babaioff et al., 2012), several key properties make the sale of data crucially different from the sale of physical goods:

- *Ex ante unverifiable*: It is hard to verify beforehand whether the data is suitable for training an ML model or not (Agarwal et al., 2019). In practice, even trained on the same data, two models of different design can yield different performances. It is typical that the true performance (i.e., the usefulness of data) of the ML model can be revealed only after deploying (possibly some portion of) data into the ML model for training.
- *Free duplication*: Data can be freely duplicated with negligible marginal cost. Therefore, once a machine learner accesses some portion of the data (e.g., to partially verify its quality), the value of these data is immediately lost since the learner can keep a copy of that portion of data.

Challenges. The above two properties post new challenges to the mechanism design of selling data to a machine learner. The ex ante unverifiable property implies that the usefulness of data (i.e., quality of data) depends on both the ML model design and the data set (e.g., size, distribution), which are controlled by the learner and the seller respectively. Therefore, neither the seller nor the learner knows the true quality of data in advance, due to i) The machine learner cannot access the seller's data due to the seller's concern of free duplication and thus cannot evaluate the true quality of data beforehand due to the ex ante unverifiable property; ii) The data seller cannot access the learner's model (typically due to business secrecy) and thus cannot evaluate the quality of data either. This intriguing situation makes our problem fundamentally different from the recent line of work on information selling (Liu et al., 2021; Babaioff et al., 2012;

[*]α-β order [1]Department of Computer Science, City University of Hong Kong, Hong Kong, China [2]Department of Computer Science, University of Chicago, Chicago, Illinois, USA (work done while this author is at UVA). Correspondence to: Junjie Chen <junjchen9-c@my.cityu.edu.hk>, Minming Li <minming.li@cityu.edu.hk>, Haifeng Xu <haifengxu@uchicago.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

Chen et al., 2020; Cai & Velegkas, 2021), in which the seller is assumed to have access to the state of the nature, i.e., the quality of the data.

To tackle the above challenges, we employ ideas from signaling (Kamenica & Gentzkow, 2011; Dughmi, 2017), and propose *Data Pricing via Costly Signaling*, so that communication is allowed between the seller and the machine learner. The proposed *Pricing via Costly Signaling* scheme consists of two steps:

- *Costly signaling step*: The seller gives away a subset of data to the learner for training the model. Both the seller and buyer then simultaneously learn a *preliminary* model accuracy, based on which they can update their belief about the underlying data quality. We view this preliminary accuracy as an informative and random *signal* of the true data quality. Notably, such informative signals can only be generated by giving away a subset of data, which immediately loses its sales value and thus reduces the available amount of data for sale. This is why it is *costly* signaling for the seller.

- *Pricing step*: Based on the observed preliminary accuracy and the amount of data given away, the seller prices the *remaining* data to maximize her revenue.

It is important to note that both parties will be able to observe preliminary outcome since each party provides essential components for the training, i.e., the partial data and the model respectively.

Two key challenges arise in the design of Pricing via Costly Signaling. Firstly, the critical part of the design is to determine the subset of data to be shared, which should balance the loss of sales value and estimation of quality. If the shared subset reveals a poor accuracy leading to poor estimation of the quality of data, the machine learner may be reluctant to pay a high price. If the shared subset reveals a high accuracy, the seller will suffer a severe loss of sales value leading to low revenue. Secondly, solving the Pricing via Costly Signaling scheme turns out to be quite challenging. Although we make use of the concept of signaling, the signaling scheme in our setup is different from previous ones in a crucial way. Traditional signaling schemes allow the sender to design signals arbitrarily correlated with the state of nature. Such signaling schemes are "smooth" so that they form a polytope conveniently characterized by linear constraints. However, in our proposed scheme, the seller cannot reveal arbitrary signals but instead is constrained to only give away a *subset* of data for sharing. This gives rise to a combinatorial signaling space with exponentially many choices because different subsets may introduce a different distribution from which the signal (i.e., the preliminary accuracy) is sampled.

Results and Techniques. We consider the problem under two different situations: homogeneous data and heterogeneous data. On one hand, the reason for considering homogeneous data is that in typical ML models, data points are usually assumed to be independent and identically distributed (i.i.d.) (Bishop, 2006). On the other hand, the sale of heterogeneous data is motivated by selling multiple different data sets, each viewed as a data "point". For instance, in a typical ML setting, features of data are heterogeneous since each feature describes one aspect of data and contains different amount of information. Therefore, the problem of selling features can be viewed as a problem of selling heterogeneous data.

In the case of homogeneous data, we assume that the model's performance (i.e., accuracy) depends on both the quality and the quantity of data. After sharing some amount of data, the optimal price for the *remaining* data is determined by a posted price mechanism, in which the price is dependent on the machine learner's private type. Interestingly, we observe that when the valuation function has certain separable forms, the optimal mechanism is to sell the entire data set without the need of costly signaling at the first. The key insight revealed in our proof is that with separable-form valuation functions, the effects of the private type and the quantity of data on the revenue can be "decoupled", leading to a more tractable design problem. In addition, we study the mechanism design for the case where the two parties may hold different prior beliefs over the quality, due to different perceptions about the data quality before the trade. We show a *revenue smoothness* type of result with respect to the prior difference; that is, even the seller may mis-perceive the learner's true prior belief, her revenue will not suffer much as long as their priors do not differ much.

Next, we move to the problem of selling heterogeneous data (e.g., the sales of features). This deviation from the homogeneous case turns out to lead to a much more difficult optimization task. Specifically, while the homogeneous case is polynomial solvable, we prove that it is NP-hard to obtain a $\frac{e}{1+e} + o(1)$ approximation for the heterogeneous case. Our proof is through a reduction from an interesting and novel combinatorial problem, which we coined as *Column Subset Selection* and may be of independent interest. Finally, we can obtain a simple approximation algorithm for this problem which directly sells the entire dataset without communications, whose approximation ratio depends on the number of the learner's private types.

Related Works. The two recent works most relevant to ours are the data marketplace design (Agarwal et al., 2019) and the model-based pricing (Chen et al., 2019) for ML models. (Agarwal et al., 2019) designed a data marketplace where multiple sellers supply data for sale and multiple

buyers come with their own ML models dynamically. (Chen et al., 2019) proposed a model-based pricing market where a trained ML model is sold, depending on the buyer's interest. Both of them assume that the marketplace or the seller can access both the ML model and data. However, in our model, the seller does not know the ML model and has to offer a small amount of data at some cost to the buyer in order to learn the buyer's ML model properties. Our work is also related to the sale of information, which can be partially revealed through signaling schemes. As far as we know, (Babaioff et al., 2012) is the first to study this problem computationally, followed by a series of recent works (Chen et al., 2020; Liu et al., 2021; Cai & Velegkas, 2021). Within this line of research, the most relevant to ours is (Zheng & Chen, 2021); they also consider a two-round mechanism, which reveals some partial information for free first to change the buyer's belief and then sell all the information at some price. The key difference between our work and this literature of selling information is that the way our seller can signal information is significantly and realistically constrained — the seller in our model can only reveal a subset of data points to signal the quality of the data. A few other works examines issues such as buyer externalities on data marketplaces (Mehta et al., 2019; Agarwal et al., 2020). Such externality does not present in our basic model since we consider the sale of data to only one machine learner. Finally, our research is also related to signaling, also known as persuasion or information design (Kamenica & Gentzkow, 2011; Dughmi & Xu, 2019; 2017), during which the sender sends a signal to receivers in order to influence receivers' decisions.

2. Preliminaries

In this paper, we assume that there is a monopoly seller who privately holds a data set $\mathcal{D}$ for sale, and a machine learner who would like to purchase $\mathcal{D}$ to train his private ML model. As we discussed in Section 1, the usefulness of data, measured with the *quality of data* q, is known to neither the seller nor the machine learner at the beginning of sales process, i.e., no one knows more than the other about q. Hence, following the standard assumption in information design (Kamenica & Gentzkow, 2011; Dughmi, 2017), we assume that the seller and the learner share the same prior belief $\mu(q)$, which is commonly known. This is reasonable because in practice, the seller and the machine learner may use publicly available resources (e.g., public available ML models on the Internet for similar task and the description of the dataset) to estimate the q. Later in Section 3.2, we will look at the robust mechanism design where two parties may privately hold different prior beliefs.

Throughout the paper, we use $\mathcal{D}$ to represent a set of "data point"s for sale. The reader may also interpret each data point $\mathcal{D}$ as a set of data or a feature. The whole selling process is depicted in Figure 1. Firstly, to persuade the machine learner to pay higher price to purchase and get a better estimation of data quality q, the seller shares with the learner a subset of the data points for free trial. We denote this subset of data points as $\mathcal{T} \subseteq \mathcal{D}$. With the set $\mathcal{T}$ of data points provided, the machine learner can train his ML model which then outputs a preliminary model accuracy r_s, that can be simultaneously observed by the two parties. Importantly, to observe the model accuracy and consequently the learner's utility, the seller does not need to know the learner's model details but only needs to observe the predictions on the testing data which typically is public. It is natural to assume that the seller has the right to observe these predictions, which are all she needs to calculate the preliminary accuracy r_s. After that, the seller posts a price $p_{r_s,\mathcal{T}}$ for the remainder of the data. Depending on the subset $\mathcal{T}$ shared and the observed accuracy r_s, the price $p_{r_s,\mathcal{T}}$ varies. Conditioning on the posted price $p_{r_s,\mathcal{T}}$, the machine learner determines whether to pay the price to purchase the remainder of data or decline the offer.

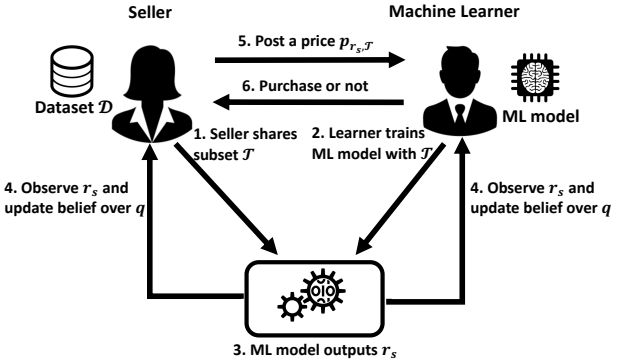

Figure 1. The whole selling process.

(Lei et al., 2019) shows that the model accuracy depends on the design of the neural network (i.e., the ML model) and the training data (e.g., the size of training set or the matching degree between distributions of training and testing set), both of which are captured by our defined q. We abstract such correlation and assume there exists a commonly known distribution $\lambda(r|q,\mathcal{T})$ for accuracy r depending on the data quality q and the set $\mathcal{T}$ of data points. In the homogeneous case where two sets are considered the same as long as they contains the same number of data points, there are only a polynomial number of different distributions $\lambda(r|q,\mathcal{T})$ because $\mathcal{D}$ has a polynomial number of data points. However, in the heterogeneous case, the exponential number of choices of subset $\mathcal{T}$ may lead to exponential number of different distributions. We assume $\lambda(r=0|q,\emptyset)=1$ if $\mathcal{T}=\emptyset$. q is discrete and finite.

The machine learner has a commonly known non-negative

valuation function $u(r, b)$ dependent on both the model's accuracy r and his private type b. The private type b abstracts the private information the learner holds, e.g., the unit price that he would pay for one percent accuracy. We assume b is discrete and finite, and follows a commonly known distribution $\varphi(b)$. $u(r, b)$ is non-decreasing in r and b. We assume $\forall b, u(0, b) = 0$ if accuracy $r = 0$.

After observing accuracy r_s, the seller and the learner update the estimation of q with Bayes rule

$$\mathbf{Pr}\left(q|r_s, \mathcal{T}\right) = \frac{\lambda(r_s|q, \mathcal{T})\mu(q)}{\sum_q \lambda(r_s|q, \mathcal{T})\mu(q)}. \tag{1}$$

Given the whole set of data points $\mathcal{D}$ for training, the model accuracy r_m is inferred by $\sum_q \lambda(r_m|q, \mathcal{D})\mathbf{Pr}\left(q|r_s, \mathcal{T}\right)$. Then the learner's expected utility obtained from the whole set $\mathcal{D}$, given the private type b, is computed as

$$\begin{aligned}&\mathbb{E}[u(r_m, b)|r_s, \mathcal{T}, \mathcal{D}] \\ &\quad= \sum_{r_m} u(r_m, b) \sum_q \lambda(r_m|q, \mathcal{D})\mathbf{Pr}\left(q|r_s, \mathcal{T}\right).\end{aligned} \tag{2}$$

After sharing set $\mathcal{T}$ of data points, the seller will immediately lose some amount of sales value, due to the free duplication property. The remaining utility is computed as

$$G(r_s, \mathcal{T}, b) = \mathbb{E}[u(r_m, b)|r_s, \mathcal{T}, \mathcal{D}] - u(r_s, b). \tag{3}$$

The learner pays the price to purchase only if his expected remaining utility $G(r_s, \mathcal{T}, b)$ is larger than the posted price $p_{r_s,\mathcal{T}}$. This can be formulated as a posted price mechanism. Hence, the expected revenue of the seller is computed as

$$\sum_{r_s} \mathbf{Pr}\left(r_s|\mathcal{T}\right) \sum_b \varphi(b) \cdot p_{r_s,\mathcal{T}} \cdot \mathbf{1}\Big\{G(r_s, \mathcal{T}, b) \geq p_{r_s,\mathcal{T}}\Big\}, \tag{4}$$

where $\mathbf{Pr}\left(r_s|\mathcal{T}\right)$ is the seller's belief of observing accuracy r_s after sharing set $\mathcal{T}$ with the learner, computed as $\mathbf{Pr}\left(r_s|\mathcal{T}\right) = \sum_q \lambda(r_s|q, \mathcal{T})\mu(q)$ and $\mathbf{1}\{\cdot\}$ is an indicator function. For convenience, the accuracy r, r_s and r_m are assumed to be discrete.[1] We refer the reader to Appendix K for a summary of notations.

To maximize her revenue formulated in (4), the seller needs to determine which set $\mathcal{T}$ of data points to be shared and the corresponding price menu $\{p_{r_s,\mathcal{T}}\}_{r_s}$. Note that when $\mathcal{T} = \emptyset$, (4) computes the revenue of selling the entire dataset without sharing any data (i.e., without costly signaling). The formulations of the seller's revenue for the homogeneous data case and the heterogeneous data case only differ in the realizations of distribution $\lambda(r|q, \mathcal{T})$, however, leading to two completely different problems.

[1] Accuracy values are naturally discrete any way since a learner can only evaluate the accuracy on a discrete data set (e.g., with size n) which can only lead to discrete accuracy estimation of the form $\frac{k}{n}$ for some integer k.

3. Pricing Homogeneous Data

In this section, we present the details of mechanism design for pricing homogeneous data. The homogeneity assumption assumes data points are i.i.d., which means that each data point contributes equally to the performance of the ML model. Therefore, the model accuracy distribution is then realized as

$$\lambda(r|q, \mathcal{T}) = \lambda(r|q, t), \tag{5}$$

where t denotes the shared *quantity of data* which is computed as $t = \frac{|\mathcal{T}|}{|\mathcal{D}|}$. By definition, we can see that t is finite and discrete and $0 \leq t \leq 1$. Sharing larger quantity of data may lead to more accurate estimate of q but at the same time, larger loss of sales value. Thus, the objective of mechanism design is to find an optimal quantity of data t^* such that the revenue is maximized while losing as little as possible sales value caused by sharing data. Note that though $\lambda(r|q, t)$ should be modeled carefully in practice, our results hold generally and are oblivious to its choice.

In the following, we divide the discussion into two parts: Section 3.1 considers the case where the seller and the machine learner have the common prior beliefs over q, i.e., $\mu(q)$. Afterwards, we will consider a harder case where the prior beliefs of two parties may differ. We include the discussion of this case in Section 3.2 and consider the robust mechanism design problem.

3.1. Mechanism Design for the Common Prior Case

We use the model accuracy distribution defined in (5) to realize the general model in Section 2. Specifically, the remaining utility defined in (3) becomes

$$G(r_s, t, b) = \mathbb{E}[u(r_m, b)|r_s, t] - u(r_s, b), \tag{6}$$

where the learner's expected utility is $\mathbb{E}[u(r_m, b)|r_s, t] = \sum_{r_m} u(r_m, b) \sum_q \lambda(r_m|q, 1)\mathbf{Pr}\left(q|r_s, t\right)$. Then, we compute the seller's expected revenue similarly as in (4)

$$\sum_{r_s} \mathbf{Pr}\left(r_s|t\right) \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{G(r_s, t, b) \geq p_{r_s,t}\Big\}, \tag{7}$$

where $\mathbf{Pr}\left(r_s|t\right) = \sum_q \lambda(r_s|q, t)\mu(q)$. To maximize the revenue, the seller needs to determine the optimal quantity of data t^* and the price menu $\{p_{r_s,t^*}\}_{r_s}$. Because t and r_s are discrete and finite, the optimal mechanism can be computed by enumeration (see Appendix B).

Before moving forward, we define a quantity $R^{\mathcal{D}}(t)$ as below

$$\begin{aligned}R^{\mathcal{D}}(t) = \max_{\{p_{r_s,t}\}_{r_s}} &\sum_{r_s} \mathbf{Pr}\left(r_s|t\right) \\ &\cdot \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{\mathbb{E}[u(r_m, b)|r_s, t] \geq p_{r_s,t}\Big\}.\end{aligned} \tag{8}$$

Compared with (7), the formulation of $R^{\mathcal{D}}(t)$ removes the sales value loss term $u(r_s, b)$ within the indicator function. In another word, $R^{\mathcal{D}}(t)$ assumes sharing partial data helps with the estimation of q but causes no loss of sales value. Hence, $R^{\mathcal{D}}(t)$ can be naturally considered as an upper bound of the seller's revenue when she shares t quantity of data.

The seller may want to quickly get a sense of whether she can make profit from this trade before enumerating t and p to maximize (7). Thus, it is very natural to ask these two questions: 1) *Is it always profitable to share data?* 2) *Under what conditions will sharing data be profitable?* Surprisingly, we find that 1) in some cases, it is optimal for the seller to sell the entire dataset directly (Theorem 3.1), and 2) the seller probably makes more revenue from sharing data if more training data gives better performance and the learner has rather small valuation for low-accuracy model. Next, we answer these two questions separately.

First Question. One kind of widely-used valuation functions is linear function, e.g., $u(r, b) = r \times b$ which can be interpreted as that the buyer would pay b for 1 unit model accuracy and $r \times b$ for accuracy r in total. We say such a linear function $u(r, b) = r \times b$ has a separable form. *Separability* means that a function can be divided into several parts, where each part depends only on one variable. We give the formal definition below.

Definition 3.1 (Separable valuation functions.) *The valuation function $u(r, b)$ is said to be multiplicatively [additively] separable if it has the form $u(r, b) = f(b)h(r)$ [$u(r, b) = f(b) + h(r)$] where $f(b)$ and $h(r)$ are any two functions of b and r respectively.*

In the following, Theorem 3.1 shows that if the valuation function $u(r, b)$ has certain separable forms, sharing data with the machine learner is not profitable to the seller.

Theorem 3.1 *If valuation function $u(r, b)$ is multiplicatively separable, then the best strategy for the seller is to sell the entire dataset directly without sharing any data. It also holds for additively separable function if $\lambda(0|q, t > 0) = 0$.*

Note that the condition $\lambda(0|q, t > 0) = 0$ ensures that training on the shared data surely returns a positive preliminary accuracy, i.e., both parties will not observe accuracy $r_s = 0$. Clearly, since the linear valuation function $u(r, b) = r \times b$ is multiplicatively separable, it is better for the seller not to share any data by Theorem 3.1. We give a sketch of proof below. The full proof is in Appendix C.

Proof of Theorem 3.1. (*sketch*) The first part of the proof lies in the key observation that for a multiplicatively separable valuation function, $R^{\mathcal{D}}(t)$ no longer depends on the shared quantity t. In another word, the value of $R^{\mathcal{D}}(t)$ is the same for all t including $t = 0$. Since $R^{\mathcal{D}}(t)$ is an upper bound of the seller's revenue defined in (7) and sharing data (i.e., $t > 0$) causes loss of sales value, the seller obtains less revenue than selling the entire dataset directly without sharing. The second part of the proof for the additively separable valuation function shares a similar idea, but in this case, we observe that the maximum revenue from (7) for $t > 0$ does not depend on the choice of private type b any more. □

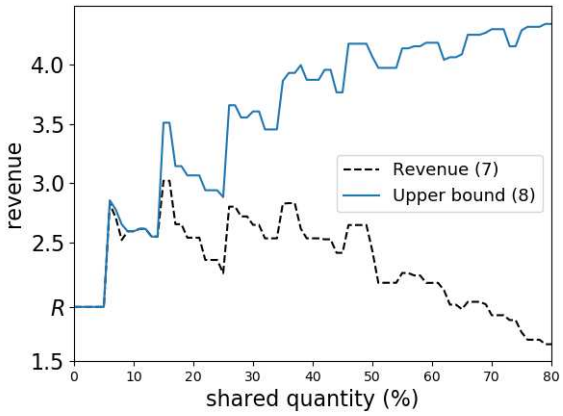

Figure 2. Revenue change with respect to quantity t. Point R is the revenue obtained by selling data without sharing any set.

Note that Theorem 3.1 holds only for separable valuation functions. Our illustrative Example 3.3 shows that the optimal mechanism general will share a subset of data with the machine learner before selling the entire data set.

Second Question. To answer the second question, we first show the following proposition. (Proof in Appendix D.)

Proposition 3.2 *The upper bound $\mathcal{R}^{\mathcal{D}}(t)$ of revenue from sharing data with the machine learner is larger than the revenue from selling the entire dataset directly without sharing any data. Formally, for any $t > 0$, we have*

$$\max_{p} \sum_{b} \varphi(b) \cdot p \cdot \mathbf{1}\Big\{ \sum_{r_m} u(r_m, b) \sum_{q} \lambda(r_m|q, 1)\mu(q) \geq p \Big\} \leq R^{\mathcal{D}}(t).$$

Remark 3.1 *Note that $\mathcal{R}^{D}(0)$ is exactly the revenue obtained from selling entire set directly. By Proposition 3.2, we know that sharing data will increase the upper bound $\mathcal{R}^{D}(t)$ over $\mathcal{R}^{D}(0)$. Hence, if the sales value loss caused by sharing data is less than the increase of the upper bound, then sharing partial data with the learner will benefit the seller's revenue.*

Finally, we use Example 3.3 to give a sense about the above finding and how the revenue (i.e., (7)) and the upper bound (i.e., $\mathcal{R}^{\mathcal{D}}(t)$) change w.r.t. the shared quantity t. The curves of Example 3.3 are plotted in Figure 2. It is interesting to note that two curves are non-monotone, non-concave and even non-smooth.

Example 3.3 *Let $t = 0\%, 1\%, \dots 100\%$ be the quantity of data. Let $r \in \{0, 1, 2, \dots, 10\}$ represent $0\%, 10\%, \dots 100\%$ accuracy, $q \in \{0, 1, 2, \dots, 10\}$ and private type $b \in \{1, 2, \dots, 10\}$. According to the above characterization, let the valuation function be*

$$u(r, b) = \begin{cases} 0, & r + b <= 10 \\ 1000, & r = b = 10 \\ 10, & otherwise. \end{cases}$$

The prior belief $\mu(q)$ over q is a Gaussian with standard deviation $\sigma = 3$ and mean $m = 3$. The accuracy distribution $\lambda(r|q, t)$ is also a Gaussian with $m = round(q \cdot t)$ and $\sigma = 0.1 \times (-(t - 0.5)^2 + 0.25)$. Let $\sigma = 0$ if $q = 0$. $\mu(q)$ and $\lambda(r|q, t)$ will be normalized to a probability measure.

The insight revealed from Example 3.3 is that we may require two conditions so that sharing data benefits the seller: 1) the model accuracy distribution $\lambda(r|q, t)$ is *first-order-dominance* w.r.t. t, i.e., for any $t_1 \geq t_2$ and r, $\int_0^r \lambda(x|q, t_2)dx \geq \int_0^r \lambda(x|q, t_1)dx$ which then implies $\mathbb{E}[r|q, t_1] \geq \mathbb{E}[r|q, t_2]$; and 2) $u(r, b)$ is relatively small or even negligible when the accuracy r is small, while it should be a relatively large value when r is large.

These two conditions can be justified in practice: i) First, it is natural to observe that given more training data, the ML model is more likely to output higher accuracy. Otherwise, the learner may not want to buy the remainder of data. One counter example is that for any $t_1 \geq t_2$, $\mathbb{E}[r|q, t_1] \leq \mathbb{E}[r|q, t_2]$. In this case, more training data actually harms the model's performance and sharing data will lead to worse estimation of quality. Thus, it is optimal to sell the entire set directly. ii) Second, a classifier trained on 5% of the data may achieve only 30% accuracy, which is far away from practical use. Thus, a company simply value it 0. However, it helps the estimation of how good the dataset is. On the contrary, if the company values a low-accuracy model relatively high, the seller will suffer a severe loss of sales value by sharing data and obtain less revenue. Two illustrative examples in Appendix J show that if either condition is violated, sharing data may not increase the seller's revenue.

3.2. Robust Mechanisms

We further consider a different setting where the priors over data quality of the seller and the machine learner, denoted as μ_{sl} and μ_{ml} respectively, differ from each other. We assume that the two priors μ_{sl} and μ_{ml} are private to the seller and the machine learner respectively and differ from each other for each q according to the following.

$$\forall q,\ |\mu_{sl}(q) - \mu_{ml}(q)| \leq \epsilon \mu_{sl}(q). \tag{9}$$

We will constantly use subscript sl and ml to denote the seller and the machine learner in the rest of this part.

The machine learner's estimation for q after training on shared quantity of data t is

$$\mathbf{Pr}_{ml}(q|r_s, t) = \frac{\lambda(r_s|q, t)\mu_{ml}(q)}{\sum_q \lambda(r_s|q, t)\mu_{ml}(q)}.$$

Then, his expected utility for the remaining data is computed as

$$G_{ml}(r_s, t, b) = \mathbb{E}_{ml}[u(r_m, b)|r_s, t] - u(r_s, b).$$

Since μ_{ml} is privately held by the machine learner, the seller may resort to robust mechanism design and maximize the worst-case revenue w.r.t. μ_{ml} subject to the constraint (9), defined as

$$\min_{\mu_{ml}} \sum_{r_s} \mathbf{Pr}_{sl}(r_s|t) \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{G_{ml}(r_s, t, b) \geq p_{r_s,t}\Big\}, \tag{10}$$

where $\mathbf{Pr}_{sl}(r_s|t) = \sum_q \lambda(r_s|q, t)\mu_{sl}(q)$ is the seller's prior belief over r_s after sharing t quantity of data.

In the robust mechanism design, (10) serves as a lower bound of revenue, and the seller needs to determine the optimal quantity t^* to maximize the lower bound. In fact, we have the following even stronger approximation results with respect to the true machine learner's prior μ_{ml} (full proof in Appendix E).

Theorem 3.4 *Let $\epsilon \in [0, 1]$ satisfy constraint (9). Then exists a mechanism, whose revenue is at least $\mathsf{OPT} - \frac{4\epsilon}{1-\epsilon^2}\bar{u}$, where OPT is the optimal expected revenue under the true prior belief of machine learner and $\bar{u} = \max_{r,b} u(r, b)$.*

Theorem 3.4 suggests that even though the machine learner's true prior belief is private, the revenue that the seller can obtain is still close to that obtained by knowing the true machine learner's prior, as long as their beliefs do not differ much, i.e., ϵ is small. In practice, the seller and the buyer may use publicly available resource for research to estimate q, so their priors may not differ too much.

4. Pricing Heterogeneous Data

In this section, we consider selling heterogeneous data. In typical ML settings, a given data set $X \in R^{n \times m}$ consists of n i.i.d. (homogeneous) data points where each data point is of m dimensions, i.e., m features which are heterogeneous.

For example, in a clinical dataset of n patients, each patient (data point) is assumed to be i.i.d., but each patient is described by multiple (i.e., m) heterogeneous features, such as *age, medicine, disease*, etc.. Note that in this example, one feature (e.g., the feature *age*) is a vector of all patients' age information (n dimensions) and an ML model can be trained on that feature. While selling data points is modeled as the sales of homogeneous data, the sales of *features* is naturally the selling heterogeneous data problem.

Suppose the seller possesses a set of features $\mathcal{D} = \{1, 2, 3, \dots M\}$ for training the ML model, and would like to price the feature set so as to maximize his own revenue. Similarly, we measure the usefulness of M features with the *quality vector* $q = [q_1, q_2, \dots, q_M]$ of M dimensions, where $q_i \geq 0$ is the quality for feature i. We assume there are in total N possible different quality vectors and N is finite. Given any subset of features $\mathcal{T} = \{i_1, i_2, \dots, i_k\}$, we use $q_{\mathcal{T}} = [q_{i_1}, q_{i_2}, \dots, q_{i_k}]$ as the corresponding quality subvector. To ease exposition, use $\mathcal{R}$ to denote the remainder set $\mathcal{D} \setminus \mathcal{T}$. An example with $M+1$ features of $N+1$ possible quality vectors is shown in Figure 3.

In the rest of discussion, we will consider one special case of $\lambda(r|q, \mathcal{T})$, which already leads to an NP-hard problem. Particularly, the $\lambda(r|q, \mathcal{T})$ is realized as a point distribution

$$\lambda(r|q, \mathcal{T}) = \lambda(r|q_{\mathcal{T}}) = \begin{cases} 1, & r = f(q_{\mathcal{T}}) \\ 0, & otherwise, \end{cases} \tag{11}$$

which indicates that the accuracy is exactly the quality. $f(q_{\mathcal{T}})$ computes the quality of subset $\mathcal{T}$.

Let $\mu(q)$ be the commonly known prior belief over quality vector q. Given any subset $\mathcal{T}$, we can compute $\mu(q_{\mathcal{T}})$ and conditional probability $\mu(q_{\mathcal{R}} \mid q_{\mathcal{T}})$. In the following, we use $\lambda(r|q, \mathcal{T})$ and $\lambda(r|q_{\mathcal{T}})$ interchangeably.

Similarly, the posterior estimation of q given set $\mathcal{T}$ is

$$\mathbf{Pr}\,(q|r_s, \mathcal{T}) = \frac{\lambda(r_s|q, \mathcal{T})\mu(q)}{\sum_q \lambda(r_s|q, \mathcal{T})\mu(q)}.$$

For later use, we first show that the seller's expected revenue in (4) can be further realized with (11) as (detailed derivation of (12) in Appendix F)

$$\sum_{r_s}\sum_{b} \varphi(b) \cdot p_{r_s,\mathcal{T}} \cdot \mathbf{1}\Big\{ \sum_{q|f(q_{\mathcal{T}})=r_s} \mu(q)u(f(q), b) - \sum_{q_{\mathcal{T}}|f(q_{\mathcal{T}})=r_s} \mu(q_{\mathcal{T}})u(f(q_{\mathcal{T}}), b) \geq p_{r_s,\mathcal{T}} \Big\}. \tag{12}$$

The derivation of (12) is oblivious to the choice of $f(q)$.

Remark 4.1 *Since the formulation (12) is intrinsically the same as (7), Theorem 3.1 still holds for pricing heterogeneous data, i.e., the best strategy is to sell the entire dataset without sharing for separable valuation functions.*

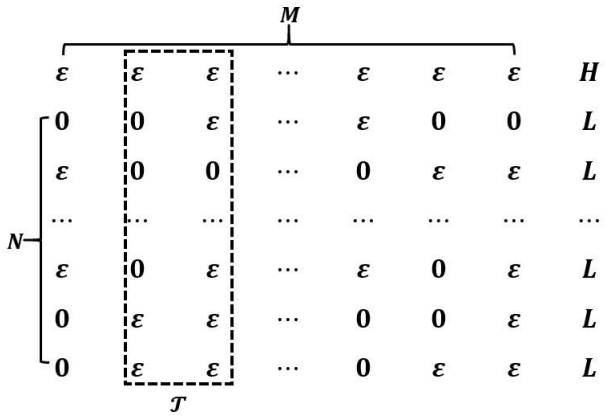

Figure 3. Constructed matrix $\mathcal{D} \in R^{(N+1)\times(M+1)}$ with 1 H-quality vector and N L-quality vectors. H and L are positive real numbers with $H > L$.

In line with (Gradwohl et al., 2021) where the allowed communication cost is limited, we impose a cardinality constraint $|\mathcal{T}| \leq T$. This is reasonable because each feature may be of high dimensions (i.e., the number of data points is large) and thus the training of the ML model is time-consuming, e.g., training a BERT model (Devlin et al., 2019) from scratch may take up to 4 days, so the seller may share at most T features to reduce the waiting time for preliminary accuracy.

4.1. Hardness of Optimal Mechanism

This section shows the hardness of computing the optimal mechanism. We employ the simple linear function $f(q_{\mathcal{T}}) = \sum_{i\in\mathcal{T}} q_i$ which computes the sum of entries in the given quality subvector. Note that although there may be exponential number of different choices of subset $\mathcal{T}$, by such a simple linearity assumption, there are in fact polynomial number of different values for $f(q_{\mathcal{T}})$ since the quality q_i is finite and discrete. Hence, there are only polynomial number of different distributions $\lambda(r|q, \mathcal{T})$. Nevertheless, the resulted problem is already NP-hard to solve. The main result of this section is the following theorem.

Theorem 4.1 *It is* NP*-hard to achieve* $\frac{e}{e+1} + o(1)$ *approximation to the optimal mechanism.*

The proof of Theorem 4.1 is rather involved and we show the high-level of our proof below. To ease exposition, the proof is divided into three parts: i) We first show that maximizing the revenue subject to a cardinality constraint can be simplified to a cleaner combinatorial problem, *Column Subset Selection*; ii) We prove that computing the optimal mechanism is NP-hard by showing that a binary version of *Column Subset Selection* is NP-hard.; iii) Based on a similar construction as that in the second step, we prove the

inapproximation ratio $\frac{e}{e+1}+o(1)$ by utilizing the hardness result from (Feige, 1998). See a complete proof in Appendix G.

Firstly, we notice that by considering some special construction of valuation function $u(r,b)$ (see the proof of Lemma G.1), subject to the constraint $|\mathcal{T}| \leq T$, the maximum revenue of (12) has the following cleaner formulation:

$$\sum_{r_s}\sum_{b}\varphi(b)\cdot p_{r_s,\mathcal{T}}\cdot \mathbf{1}\Big\{\sum_{q|f(q_{\mathcal{T}})=r_s}\mu(q)u(f(q),b)\geq p_{r_s,\mathcal{T}}\Big\}. \tag{13}$$

The formulation (13) can be interpreted as following: As in Figure 3, suppose that the set $\mathcal{D}$ contains $M+1$ features (columns) and there are $N+1$ possible choices of quality vectors q (rows). By selecting a subset $\mathcal{T}$ of features (i.e., set of columns in matrix $\mathcal{D}$), the $N+1$ quality vectors are divided into different groups according to the value $f(q_{\mathcal{T}})$ of the quality subvectors $q_{\mathcal{T}}$. That is, given an r_s, quality vectors with $f(q_{\mathcal{T}})=r_s$ are classified to the same group, and a price $p_{r_s,\mathcal{T}}$ needs to be determined so that the sub problem $\sum_b \varphi(b)\cdot p_{r_s,\mathcal{T}}\cdot \mathbf{1}\{\sum_{q|f(q_{\mathcal{T}})=r_s}\mu(q)u(f(q),b)\geq p_{r_s,\mathcal{T}}\}$ is maximized. If the valuation $u(f(q),b)$ is different for different quality vectors q, the maximum value of (13) is achieved only when all the quality vectors q are separated from each other. The maximization of (13) subject to the constraint is intrinsically equivalent to the following *Column Subset Selection* problem, which we find is of independent interest.

COLUMN SUBSET SELECTION

Input: n arbitrary nonnegative vectors $\mathbf{x}^1,\mathbf{x}^2,\ldots,\mathbf{x}^n$ of m-dimension.

Output: find a minimum-size set of entries $\mathcal{E}\subseteq[m]$ to distinguish all vectors according to the sum of entries, i.e., $\sum_{i\in\mathcal{E}}\mathbf{x}_i^1\neq\sum_{i\in\mathcal{E}}\mathbf{x}_i^2\neq\ldots\neq\sum_{i\in\mathcal{E}}\mathbf{x}_i^n$.

The fundamental challenge in our costly signaling problem is essentially the same as the *Column Subset Selection* problem above: how to select the most effective — in the sense of good quality estimation but small loss of sales value caused by sharing data — subset of features (i.e., columns) is hard.

Secondly, we prove the hardness of costly signaling, i.e., the maximization of (12), by proving the hardness of a binary version of the *Column Subset Selection* problem, which distinguishes only one vector from others [2]. As in Figure 3, the constructed set $\mathcal{D}$ consists of 1 H-quality vector and N L-quality vectors, where the H-quality vector has value H in the last entry and ϵ in other M entries, and the L-quality vector has value L in the last entry while 0 and ϵ in its first M entries. We can show that the maximum value of problem (13) is achieved by separating the H-quality vector from all L-quality vectors. The NP-hardness is proved by a reduction from the set cover problem (Cormen et al., 2009): Given a ground set $\mathcal{U}=\{1,2,\ldots,N\}$, a collection of M subsets of $\mathcal{U}$ and an integer T, determine if there exists a collection $\mathcal{C}$ of at most T subsets so that all the elements in $\mathcal{U}$ are covered (i.e., included) in $\mathcal{C}$? Given a set cover instance, we construct one instance of (13) as follows: The ground set $\mathcal{U}$ corresponds to the N L-quality vectors. M subsets correspond to the first M columns in $\mathcal{D}$. The given integer T is the cardinality constraint.

Finally, we utilize the maximum coverage problem to prove the inapproximation ratio: Given a ground set $\mathcal{U}=\{1,2,\ldots,N\}$, a collection of M subsets of $\mathcal{U}$ and an integer T, find a collection of no more than T subsets such that the number of elements included in $\mathcal{C}$ is maximized. It is important to note that the maximum coverage and the set cover problem have the same input. (Feige, 1998) showed that given a maximum coverage instance with an integer T as the optimal value (i.e., T is the minimum number of subsets needed to cover all the elements in $\mathcal{U}$) of a set cover instance with the same input, there is no polynomial algorithm giving a coverage of size at least $(1-\frac{1}{e}+o(1))N$ where N is the number of elements in the ground set.

Let constraint T be the optimal value of a constructed set cover instance (same as the second step). Then, no polynomial algorithm can reach an approximiation ratio $1-\alpha=1-\frac{1}{e}+o(1)$ for the maximum coverage problem. In addition, let the machine learner have two private types b_1, b_2:

- Set probabilities $\varphi(b_1)=1-\frac{1}{\alpha N}$ and $\varphi(b_2)=\frac{1}{\alpha N}$, respectively. $b_1<b_2$.
- Let $u(L,b_1)=1-\frac{1}{N}, u(L,b_2)=1$ and $u(H,b_1)=1$, $u(H,b_2)=(\alpha N)^2$.

The optimal revenue is achieved by separating the H-quality vector from all L-quality vectors, which is $\max_{p_H}\sum_b\varphi(b)\cdot p_H\cdot\mathbf{1}\{u(H,b)\geq p_H\}+\max_{p_L}\sum_b\varphi(b)\cdot p_L\cdot\mathbf{1}\{Nu(L,b)\geq p_L\}=\alpha N+N-1$. By utilizing the hardness result from (Feige, 1998), we can have that the revenue obtained by any polynomial-time algorithm is upper bounded by $N+\alpha$. Given $\alpha=\frac{1}{e}-o(1)$, we have ratio $\frac{N+\alpha}{\alpha N+N-1}=\frac{N}{\alpha N+N}+o(1)=\frac{e}{e+1}+o(1)$.

4.2. Approximation Algorithm

We seek an approximation algorithm that can compute the mechanism in a fast way for pricing the heterogeneous data. One natural idea is to greedily select the feature that gives the largest increase of revenue. However, in the following, we give some negative observations about this natural idea.

[2] We thank an anonymous reviewer for suggesting an idea to modify our reduction in order to prove the hardness of the general *Column Subset Selection* (CSS) problem. Though the hardness of the general CSS problem is not needed for our result, we believe it is indeed an interesting result, and thus prove it in Appendix I.

The proof is in Appendix H.

Observation 4.2 *In general, the revenue (12) is not a submodular function regarding the set $\mathcal{T}$. Additionally, in terms of the increase of revenue, greedy algorithm (i.e., greedily select feature that gives the largest increase of revenue) may give arbitrarily bad approximation.*

Despite the above negative results, we show that a simple approximation algorithm which sells the entire dataset directly according to the common prior gives a relatively tight approximation if the number of machine learner's private types is small.

Theorem 4.3 *Oblivious to the choice of $f(q)$, by selling the entire dataset directly without sharing any feature, at least $\frac{1}{k}\mathsf{OPT}$ revenue can be obtained, where k is the number of private types and OPT is the optimal revenue.*

The key idea of the proof is to compare the revenue of selling data directly according to the prior belief and the upper bound of (12), i.e., $\max_{p_q} \sum_q \sum_b \varphi(b) \cdot p_q \cdot \mathbf{1}\{\mu(q)u(f(q), b) \geq p_q\}$, which is obtained by separating all quality vectors from each other and removing sales value loss term. When $k = 2$, this simple algorithm can achieve a ratio $\frac{1}{2}$, which is close to the hardness results $\frac{e}{e+1} \approx 0.7$. We remark that some real-world applications do fall within the regime of small k. For example, when selling data of human face pictures for recognition, the buyers typically come from three types: researchers, governmental agencies and companies.

5. Conclusions and Future Works

We take the first step to consider the problem of selling data to a machine learner. A *Pricing via Costly Signaling* scheme is proposed for the homogeneous data and heterogeneous data cases. We believe the problem of selling data to a machine learner is of great significance to both theory research and practical application. A few open questions are left, which we believe deserve further investigation:

- Is there a better approximation algorithm for maximizing (12) to achieve higher revenue?
- Besides selling data by pricing via costly signaling, is there a better mechanism to sell data under the same setting? It would also be interesting to consider selling data to multiple learners (or companies).

Acknowledgment. Haifeng Xu is supported by the NSF grant CCF-2132506. Minming Li is supported by a grant from Research Grants Council of Hong Kong SAR, China (Project No. CityU 11205619).

References

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

Agarwal, A., Dahleh, M., Horel, T., and Rui, M. Towards data auctions with externalities. *arXiv preprint arXiv:2003.08345*, 2020.

Babaioff, M., Kleinberg, R., and Paes Leme, R. Optimal mechanisms for selling information. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pp. 92–109, 2012.

Bergemann, D., Bonatti, A., and Gan, T. The economics of social data. *RAND Journal of Economics (forthcoming)*, 2019.

Bishop, C. M. Pattern recognition. *Machine learning*, 128 (9), 2006.

Cai, Y. and Velegkas, G. How to sell information optimally: An algorithmic study. In *Proceedings of the12th Innovations in Theoretical Computer Science Conference*, volume 185, 2021.

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

Chen, Y., Xu, H., and Zheng, S. Selling information through consulting. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2412–2431. SIAM, 2020.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to algorithms*. MIT press, 2009.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

Dughmi, S. Algorithmic information structure design: a survey. *ACM SIGecom Exchanges*, 15(2):2–24, 2017.

Dughmi, S. and Xu, H. Algorithmic persuasion with no externalities. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pp. 351–368, 2017.

Dughmi, S. and Xu, H. Algorithmic bayesian persuasion. *SIAM Journal on Computing*, (0):STOC16–68, 2019.

Feige, U. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

Gradwohl, R., Hahn, N., Hoefer, M., and Smorodinsky, R. Algorithms for persuasion with limited communication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 637–652. SIAM, 2021.

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

Lei, S., Zhang, H., Wang, K., and Su, Z. How training data affect the accuracy and robustness of neural networks for image classification, 2019. URL https://openreview.net/forum?id=HklKWhC5F7.

Liu, S., Shen, W., and Xu, H. Optimal pricing of information. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pp. 693–693, 2021.

Mehta, S., Dawande, M., Janakiraman, G., and Mookerjee, V. How to sell a dataset? pricing policies for data monetization. In *20th ACM Conference on Economics and Computation, EC*, pp. 679, 2019.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., and Iyengar, S. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.

Zheng, S. and Chen, Y. Optimal advertising for information products. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pp. 888–906, 2021.

A. Preliminaries

Given a posted-price mechanism formulated as

$$\max_{p} \sum_{b} \varphi(b) \cdot p \cdot \mathbf{1}\{f(b) \geq p\},$$

where b is a private type, $\varphi(b)$ is the probability of b and $f(b) \geq 0$ only depends on b. We have the following observations:

Observation A.1 *The optimal price p^* must be $p^* = f(b^*)$ for some private type b^*. b^* is called optimal type.*

Observation A.2 *Denote $\mathbf{Pr}\left(f(b) < p\right)$ as the probability of not purchasing the item at some price p, i.e., $\mathbf{Pr}\left(f(b) < p\right) = \sum_{b:f(b)<p} \varphi(b)$. If $f(b)$ is increasing in b, $\mathbf{Pr}\left(f(b) < p^*\right)$ can be computed with the cumulative probability function Φ of φ, as $\mathbf{Pr}\left(f(b) < p^*\right) = \Phi(b^*) = \sum_{b<b^*} \varphi(b)$.*

B. Solution to Maximizing (7)

Lemma B.1 *The optimal quantity t^* and the price menu $\{p_{r_s,t^*}\}_{r_s}$ to maximizing (7) can be computed in polynomial time.*

Proof: Note that the problem of maximizing (7) is

$$\max_{\{p_{r_s,t}\}_{r_s},\, t} \quad \sum_{r_s} \mathbf{Pr}\left(r_s|t\right) \sum_{b} \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{\mathbb{E}[u(r_m,b)|r_s,t] - u(r_s,b) \geq p_{r_s,t}\Big\}. \tag{14}$$

Given quantity t and the observed accuracy r_s, maximizing the seller's revenue by solving (14) is equivalent to finding a $p_{r_s,t}$ such that $\sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{\mathbb{E}[u(r_m,b)|r_s,t] - u(r_s,b) \geq p_{r_s,t}\Big\}$ is maximized. Now, the problem turns into a posted price mechanism design problem. Since the machine learner's private type b is finite and discrete, the optimal price $p^*_{r_s,t}$ is equal to $f(b^*) = \mathbb{E}[u(r_m,b^*)|r_s,t] - u(r_s,b^*)$ for some b^*, and can be determined by enumeration in polynomial time.

Once the price $p^*_{r_s,t}$ is given, the expected revenue conditioning on t and r_s is also determined, computed as $Rev(r_s,t) = \Big\{1 - \mathbf{Pr}\left(f(b) < p^*_{r_s,t}\right)\Big\} \cdot p^*_{r_s,t}$. It is possible that the utility of the remaining data $\mathbb{E}[u(r_m,b^*)|r_s,t] - u(r_s,b^*) < 0$. Therefore, the revenue should be $\max\big\{Rev(r_s,t), 0\big\}$, given t and r_s.

To find the optimal t^*, we can enumerate all t because t is finite. □

C. Proof of Theorem 3.1

We divide the proof into two parts for multiplicatively and additively separable valuation function respectively. First, we have the following key lemma for our proof.

Lemma C.1 *Given two posted price mechanisms formulated as $\max_p \sum_b \varphi(b) \cdot p \cdot \mathbf{1}\{f(b) \geq p\}$ and $\max_p \sum_b \varphi(b) \cdot p \cdot \mathbf{1}\{c \cdot f(b) \geq p\}$ respectively, where $\varphi(b)$ is the probability of b, $c \geq 0$ is some constant and $f(b) \geq 0$ is increasing in b, there exists one optimal type b^* (see Observation A.1) giving the optimal revenues for two mechanisms as $(1 - \Phi(b^*))f(b^*)$ and $(1 - \Phi(b^*)) \cdot cf(b^*)$, where Φ is the cumulative probability function of φ (see Observation A.2).*

Proof: We know that the optimal solution for the fisrt posted price mechanism is $(1 - \Phi(b^*))f(b^*)$ as $f(b) \geq 0$ is increasing in b. Because $\max_p \sum_b \varphi(b) \cdot p \cdot \mathbf{1}\{c \cdot f(b) \geq p\} = c \cdot \max_p \sum_b \varphi(b) \cdot p \cdot \mathbf{1}\{f(b) \geq p\}$, there must exist one optimal b^* for two mechanisms such that the optimal revenues are as in the claim. □

First, we prove that the theorem holds for the multiplicatively separable valuation function $u(r,b) = f(b)h(r)$.

Lemma C.2 *Consider the upper bound of (7), i.e., $R^{\mathcal{D}}(t)$ defined in (8), and we formulate the problem of maximizing the upper bound as*

$$\max_{\{p_{r_s,t}\}_{r_s},\, t} \quad \sum_{r_s} \mathbf{Pr}\left(r_s|t\right) \sum_{b} \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{\mathbb{E}[u(r_m,b)|r_s,t] \geq p_{r_s,t}\Big\}. \tag{15}$$

For multiplicatively separable valuation function $u(r,b) = f(b)h(r)$ where $f(b)$ and $h(r)$ are two functions depending only on b or r, the upper bound $R^{\mathcal{D}}(t)$ is the same for different quantity t.

Proof: Note that similar as Lemma B.1, the maximum of $R^{\mathcal{D}}(t)$ must be obtained at some t^*. Hence, we can assume a data quantity t is given. The upper bound (15) is

$$\max_{\{p_{r_s,t}\}_{r_s}} \quad \sum_{r_s} \mathbf{Pr}\,(r_s|t) \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{\mathbb{E}[u(r_m,b)|r_s,t] \geq p_{r_s,t}\Big\} \tag{16a}$$

$$= \max_{\{p_{r_s,t}\}_{r_s}} \quad \sum_{r_s}\sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{\sum_q \sum_{r_m} u(r_m,b)\lambda(r_m|q,1)\lambda(r_s|q,t)\mu(q) \geq p_{r_s,t}\Big\}. \tag{16b}$$

The equality holds because $\mathbf{Pr}\,(r_s|t) = \sum_q \lambda(r_s|q,t)\mu(q)$ and we move $\mathbf{Pr}\,(r_s|t)$ inside the indicator function.

Note that in (16b), each r_s is associated with one posted price mechanism as below

$$\max_{p_{r_s,t}} \quad \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{\sum_q \sum_{r_m} u(r_m,b)\lambda(r_m|q,1)\lambda(r_s|q,t)\mu(q) \geq p_{r_s,t}\Big\}.$$

We now focus on the utility term within the indicator function. By replacing $u(r,b)$ with $f(b)h(r)$, we can have the utility term as $f(b) \cdot g(r_s,t)$ where

$$g(r_s,t) = \sum_q \sum_{r_m} h(r_m)\lambda(r_m|q,1)\lambda(r_s|q,t)\mu(q). \tag{17}$$

Note that conditioning on the r_s and t, $g(r_s,t)$ can be considered as a constant. Furthermore, because $u(r,b)$ is increasing in b, the optimal b^* is the same for all different r_s and t by Lemma C.1. Hence, for any given t, we have the upper bound as

$$\max_{\{p_{r_s,t}\}_{r_s}} \quad \sum_{r_s}\sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{\sum_q \sum_{r_m} u(r_m,b)\lambda(r_m|q,1)\lambda(r_s|q,t)\mu(q) >= p_{r_s,t}\Big\} \tag{18a}$$

$$= (1-\Phi(b^*)) \sum_{r_s}\sum_q\sum_{r_m} u(r_m,b^*)\lambda(r_m|q,1)\lambda(r_s|q,t)\mu(q)$$

$$= (1-\Phi(b^*)) \sum_q\sum_{r_m} u(r_m,b^*)\lambda(r_m|q,1)\sum_{r_s}\lambda(r_s|q,t)\mu(q)$$

$$= (1-\Phi(b^*)) \sum_q\sum_{r_m} u(r_m,b^*)\lambda(r_m|q,1)\mu(q)$$

$$= (1-\Phi(b^*)) f(b^*) \sum_q\sum_{r_m} h(r_m)\lambda(r_m|q,1)\mu(q), \tag{18b}$$

where $\Phi(b)$ is the same as that in Observation A.2. We can see from (18b) that the result is independent of quantity t. In another word, all the t's (including $t=0$) have the same upper bound, i.e., the maximum of (15). The lemma is proved. □

Lemma C.2 claims that with the multiplicatively separable valuation function, sharing data or not will not affect the value of upper bound, i.e., $R^{\mathcal{D}}(0) = R^{\mathcal{D}}(t), \forall t > 0$. By the free duplication property, sharing data will cause loss of sales value to the seller. Hence, we know that the best strategy in this case is not to share any data.

Corollary C.3 *If the valuation function is multiplicatively separable, the best selling strategy of (7) is to sell the entire dataset directly without sharing any data.*

Proof: We know that the maximum value of (15), which is obtained in (18b), is an upper bound of (7). Also, note that the (18b) is exactly the same as the revenue obtained from selling the entire dataset directly.

$$\max_p \quad \sum_b \varphi(b) \cdot p \cdot \mathbf{1}\{\sum_q \sum_{r_m} u(r_m,b)\lambda(r_m|q,1)\mu(q) >= p\}.$$

Since sharing partial data will cause loss to the sales value but the upper bound remains the same for $t > 0$, the seller's revenue will then decrease after sharing. □

The proof of Theorem 3.1 is done by futher considering the case $u(r,b) = f(b) + h(r)$.

Lemma C.4 *If $\lambda(0|q,t>0)=0$ and the valuation function is additively separable $u(r,b)=f(b)+h(r)$ where $f(b)$ and $h(r)$ are functions depending only on b or r, the maximum revenue of (7) is upper bounded by that from selling the entire dataset directly to the machine learner.*

Proof: If $\lambda(0|q,t>0)=0$, then the observed preliminary accuracy $r_s>0$ and $u(r_s,b)$ will have the additive separable form $f(b)+h(r_s)$ for sure. Given a fixed r_s and $t>0$, the utility term in (7) would be

$$\begin{aligned}
&\mathbb{E}[u(r_m,b)|r_s,t]-u(r_s,b)\\
=&\sum_q\sum_{r_m}u(r_m,b)\lambda(r_m|q,1)\frac{\lambda(r_s|q,t)\mu(q)}{\sum_q\lambda(r_s|q,t)\mu(q)}-u(r_s,b)\\
=&\sum_q\sum_{r_m}(f(b)+h(r_m))\lambda(r_m|q,1)\frac{\lambda(r_s|q,t)\mu(q)}{\sum_q\lambda(r_s|q,t)\mu(q)}-(f(b)+h(r_s))\\
=&\sum_q\sum_{r_m}h(r_m)\lambda(r_m|q,1)\frac{\lambda(r_s|q,t)\mu(q)}{\sum_q\lambda(r_s|q,t)\mu(q)}-h(r_s),
\end{aligned}$$

which is independent of b but only depends on r_s, that is, the utility term is the same for every private type b. In another word, for a fixed r_s, the optimal price in (7) is

$$p_{r_s,t}=\mathbb{E}[u(r_m,b_0)|r_s,t]-u(r_s,b_0), \tag{19}$$

where b_0 can be any private type and the learner will surely purchase with probability 1. For ease of exposition, let b_0 be the smallest private type and fixed for all r_s. Thus, for any $t>0$, the revenue is

$$\sum_{r_m}u(r_m,b_0)\sum_q\lambda(r_m|q,1)\mu(q)-\sum_{r_s}\mathbf{Pr}\left(r_s|t\right)u(r_s,b_0).$$

In addition, we know that when $t=0$, the revenue of selling the entire dataset directly according to the prior belief is $\max_p\ \sum_b\varphi(b)\cdot p\cdot\mathbf{1}\{\sum_{r_m}u(r_m,b)\sum_q\lambda(r_m|q,1)\mu(q)\geq p\}$. The maximum revenue of the posted price mechanism satisfies

$$\begin{aligned}
&\max_p\ \sum_b\varphi(b)\cdot p\cdot\mathbf{1}\{\sum_{r_m}u(r_m,b)\sum_q\lambda(r_m|q,1)\mu(q)\geq p\}\\
\geq&(1-\Phi(b_0))\sum_{r_m}u(r_m,b_0)\sum_q\lambda(r_m|q,1)\mu(q)\\
\geq&\sum_{r_m}u(r_m,b_0)\sum_q\lambda(r_m|q,1)\mu(q)-\sum_{r_s}\mathbf{Pr}\left(r_s|t\right)u(r_s,b_0),
\end{aligned}$$

where $\Phi(b)$ is defined similarly as in Observation A.2. The second inequality holds because $1-\Phi(b_0)=1$ (since b_0 is the smallest private type) and $\sum_{r_s}\mathbf{Pr}\left(r_s|t\right)u(r_s,b_0)\geq 0$. Hence, we can see that the best strategy for the seller in this case is to directly sell the data. Note that it is possible that $p_{r_s,t}<0$ in (19). By letting $p_{r_s,t}=\max\left\{0,\mathbb{E}[u(r_m,b_0)|r_s,t]-u(r_s,b_0)\right\}$, our analysis remains the same. □

D. Proof of Proposition 3.2

Proof: The upper bound obtained by sharing t data with the machine learner is at least

$$\begin{aligned}
&\max_{\{p_{r_s,t}\}_{r_s}} \quad \sum_{r_s}\sum_{b}\varphi(b)\cdot p_{r_s,t}\cdot \mathbf{1}\{\sum_{r_m}u(r_m,b)\sum_{q}\lambda(r_m|q,1)\lambda(r_s|q,t)\mu(q)\geq p_{r_s,t}\}\\
=&\sum_{r_s}(1-\Phi(b^*_{r_s}))\sum_{r_m}u(r_m,b^*_{r_s})\sum_{q}\lambda(r_m|q,1)\lambda(r_s|q,t)\mu(q)\\
\geq&\sum_{r_s}(1-\Phi(b^*))\sum_{r_m}u(r_m,b^*)\sum_{q}\lambda(r_m|q,1)\lambda(r_s|q,t)\mu(q)\\
=&(1-\Phi(b^*))\sum_{r_m}u(r_m,b^*)\sum_{q}\lambda(r_m|q,1)\mu(q)\\
=&\max_{p} \quad \sum_{b}\varphi(b)\cdot p\cdot \mathbf{1}\{\sum_{r_m}u(r_m,b)\sum_{q}\lambda(r_m|q,1)\mu(q)\geq p\}.
\end{aligned}$$

The first equality is from maximizing each posted price mechanism for every r_s and that $b^*_{r_s}$ is the optimal type, as in Appendix A. The second inequality holds because we use the same private type b^* for all different r_s, which may not be the optimal type for the posted price mechanism induced by some r_s. The last equality holds because we use b^* which is the optimal type of selling the entire dataset directly according to the prior belief. Thus, the proof completes. □

E. Proof of Theorem 3.4

Proof: This theorem is proved by bounding the difference between the upper bound and lower bound of (10).

For any given μ_{ml}, by the constraint $\forall q,\ |\mu_{sl}(q)-\mu_{ml}(q)|\leq\epsilon\mu_{sl}(q)$ in (9), we have

$$\begin{aligned}
&\mathbb{E}_{ml}[u(r_m,b)|r_s,t]\\
=&\sum_{q}\sum_{r_m}u(r_m,b)\lambda(r_m|q,1)\frac{\lambda(r_s|q,t)\mu_{ml}(q)}{\sum_q\lambda(r_s|q,t)\mu_{ml}(q)}\\
\geq&\sum_{q}\sum_{r_m}u(r_m,b)\lambda(r_m|q,1)\frac{\lambda(r_s|q,t)(\mu_{sl}(q)-\epsilon\mu_{sl}(q))}{\sum_q\lambda(r_s|q,t)(\mu_{sl}(q)+\epsilon\mu_{sl}(q))}\\
=&\frac{(1-\epsilon)\sum_q\lambda(r_s|q,t)\mu_{sl}(q)\sum_{r_m}u(r_m,b)\lambda(r_m|q,1)}{(1+\epsilon)\sum_q\lambda(r_s|q,t)\mu_{sl}(q)}.
\end{aligned}\tag{20}$$

Therefore, given the price $p_{r_s,t}$ for each r_s, we can derive the lower bound as

$$\begin{aligned}
&\min_{\mu_{ml}}\ \sum_{r_s}\mathbf{Pr}_{sl}\left(r_s|t\right)\sum_{b}\varphi(b)\cdot p_{r_s,t}\cdot\mathbf{1}\Big\{\mathbb{E}_{ml}[u(r_m,b)|r_s,t]-u(r_s,b)\geq p_{r_s,t}\Big\}\\
\geq&\sum_{r_s}\mathbf{Pr}_{sl}\left(r_s|t\right)\sum_{b}\varphi(b)\cdot p_{r_s,t}\cdot\mathbf{1}\Big\{\frac{(1-\epsilon)\sum_q\lambda(r_s|q,t)\mu_{sl}(q)\sum_{r_m}u(r_m,b)\lambda(r_m|q,1)}{(1+\epsilon)\sum_q\lambda(r_s|q,t)\mu_{sl}(q)}-u(r_s,b)\geq p_{r_s,t}\Big\}.
\end{aligned}$$

It is important to note that the lower bound also holds for the true machine learner's prior, which satisfies the constraints (9). We maximize the lower bound and have

$$\begin{aligned}
&\max_{\{p_{r_s,t}\}_{r_s}}\min_{\mu_{ml}}\ \sum_{r_s}\mathbf{Pr}_{sl}\left(r_s|t\right)\sum_{b}\varphi(b)\cdot p_{r_s,t}\cdot\mathbf{1}\Big\{\mathbb{E}_{ml}[u(r_m,b)|r_s,t]-u(r_s,b)\geq p_{r_s,t}\Big\}\\
\geq&\max_{\{p_{r_s,t}\}_{r_s}}\sum_{r_s}\mathbf{Pr}_{sl}\left(r_s|t\right)\sum_{b}\varphi(b)\cdot p_{r_s,t}\cdot\mathbf{1}\Big\{\frac{(1-\epsilon)\sum_q\lambda(r_s|q,t)\mu_{sl}(q)\sum_{r_m}u(r_m,b)\lambda(r_m|q,1)}{(1+\epsilon)\sum_q\lambda(r_s|q,t)\mu_{sl}(q)}-u(r_s,b)\geq p_{r_s,t}\Big\}.
\end{aligned}$$

We use $LB(r_s,b)=\frac{(1-\epsilon)\sum_q\lambda(r_s|q,t)\mu_{sl}(q)\sum_{r_m}u(r_m,b)\lambda(r_m|q,1)}{(1+\epsilon)\sum_q\lambda(r_s|q,t)\mu_{sl}(q)}-u(r_s,b)$ to denote the term within the indicator function. It is possible that $LB(r_s,b)<0$ for some r_s.

Similarly, given the price $p_{r_s,t}$ for all r_s, we have the following upper bound for any μ_{ml} satisfying the constraint (9)

$$\sum_{r_s} \mathbf{Pr}_{sl}(r_s|t) \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{\mathbb{E}_{ml}[u(r_m,b)|r_s,t] - u(r_s,b) \geq p_{r_s,t}\Big\}$$
$$\leq \sum_{r_s} \mathbf{Pr}_{sl}(r_s|t) \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{\frac{(1+\epsilon)\sum_q \lambda(r_s|q,t)\mu_{sl}(q)\sum_{r_m} u(r_m,b)\lambda(r_m|q,1)}{(1-\epsilon)\sum_q \lambda(r_s|q,t)\mu_{sl}(q)} - u(r_s,b) \geq p_{r_s,t}\Big\}.$$

We maximize the upper bound and have

$$\max_{\{p_{r_s,t}\}_{r_s}} \sum_{r_s} \mathbf{Pr}_{sl}(r_s|t) \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{\mathbb{E}_{ml}[u(r_m,b)|r_s,t] - u(r_s,b) \geq p_{r_s,t}\Big\}$$
$$\leq \max_{\{p_{r_s,t}\}_{r_s}} \sum_{r_s} \mathbf{Pr}_{sl}(r_s|t) \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\Big\{\frac{(1+\epsilon)\sum_q \lambda(r_s|q,t)\mu_{sl}(q)\sum_{r_m} u(r_m,b)\lambda(r_m|q,1)}{(1-\epsilon)\sum_q \lambda(r_s|q,t)\mu_{sl}(q)} - u(r_s,b) \geq p_{r_s,t}\Big\}.$$

We use $UB(r_s,b) = \frac{(1+\epsilon)\sum_q \lambda(r_s|q,t)\mu_{sl}(q)\sum_{r_m} u(r_m,b)\lambda(r_m|q,1)}{(1-\epsilon)\sum_q \lambda(r_s|q,t)\mu_{sl}(q)} - u(r_s,b)$ to denote the term within the indicator function in the upper bound. Aslo, note that the upper bound holds for the true machine learner's prior belief.

Furthermore, we have $UB(r_s,b) - LB(r_s,b) = \frac{4\epsilon}{1-\epsilon^2}\sum_q \sum_{r_m} u(r_m,b)\lambda(r_m|q,1)\frac{\lambda(r_s|q,t)\mu_{sl}(q)}{\sum_q \lambda(r_s|q,t)\mu_{sl}(q)} \leq \frac{4\epsilon}{1-\epsilon^2}\bar{u}$, where $\bar{u} = \max_{r,b} u(r,b)$. By subtracting the lower bound from the upper bound, we have

$$\max_{\{p_{r_s,t}\}_{r_s}} \sum_{r_s} \mathbf{Pr}_{sl}(r_s|t) \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\{UB(r_s,b) \geq p_{r_s,t}\}$$
$$- \max_{\{p_{r_s,t}\}_{r_s}} \sum_{r_s} \mathbf{Pr}_{sl}(r_s|t) \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\{LB(r_s,b) \geq p_{r_s,t}\} \tag{21a}$$
$$\leq \max_{\{p_{r_s,t}\}_{r_s}} \sum_{r_s} \mathbf{Pr}_{sl}(r_s|t) \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\{UB(r_s,b) \geq p_{r_s,t}\}$$
$$- \max_{\{p_{r_s,t}\}_{r_s}} \sum_{r_s} \mathbf{Pr}_{sl}(r_s|t) \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\{UB(r_s,b) - \frac{4\epsilon}{1-\epsilon^2}\bar{u} \geq p_{r_s,t}\}. \tag{21b}$$

In the following, we can assume for each r_s that $\max_{p_{r_s,t}} \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\{UB(r_s,b) \geq p_{r_s,t}\} > 0$. Otherwise, the revenue gap between the upper bound and the lower bound is 0, leading to the desired C-approximation.

In the (21b), there are two posted price mechanisms. After optimizing the first posted price mechanism, the gap between the upper bound and the lower bound is further upper bounded by

$$(21b) \leq \sum_{r_s} \mathbf{Pr}_{sl}(r_s|t)\, \Phi^{\geq}_{r_s}(b^*_{r_s}) UB(r_s,b^*_{r_s}) - \sum_{r_s} \mathbf{Pr}_{sl}(r_s|t)\, \Phi^{\geq}_{r_s}(b^*_{r_s})\Big(UB(r_s,b^*_{r_s}) - \frac{4\epsilon}{1-\epsilon^2}\bar{u}\Big)$$
$$= \sum_{r_s} \mathbf{Pr}_{sl}(r_s|t)\, \Phi^{\geq}_{r_s}(b^*_{r_s}) \frac{4\epsilon}{1-\epsilon^2}\bar{u}$$
$$\leq \frac{4\epsilon}{1-\epsilon^2}\bar{u},$$

where $b^*_{r_s}$ is the optimal private type maximizing $\sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\{UB(r_s,b) \geq p_{r_s,t}\}$ and $\Phi^{\geq}_{r_s}(b^*_{r_s}) = \sum_{b \in S_{r_s}} \varphi(b)$ with $S_{r_s} = \{b | UB(r_s,b) \geq UB(r_s,b^*_{r_s})\}$. Thus, when selling according to the lower bound mechanism, we obtain a C-approximate algorithm.

Mechanism: Recall that $LB(r_s,b) < 0$ can occur. The derived mechanism sets the price $p_{r_s,t} = \max\{LB(r_s,b^*_{r_s}), 0\}$ where $b^*_{r_s}$ is the optimal type for the problem

$$\max_{p_{r_s,t}} \sum_b \varphi(b) \cdot p_{r_s,t} \cdot \mathbf{1}\{LB(r_s,b) \geq p_{r_s,t}\}. \tag{22}$$

Recall that for any given accuracy r_s and private type b, $LB(r_s, b)$ lower bounds the utility that the learner computes with the true prior belief $\mu_{ml}(q)$. Hence, when the learner makes purchasing-or-not decision under $\mu_{ml}(q)$, the probability of accepting the posted price offered by the mechanism will be higher than the probability obtained in (22), leading to higher revenue. The optimal t^* can be obtained by enumeration as Lemma B.1.

In summary, the mechanism works as below:

1. The seller shares t^* quantity of data with the machine learner;
2. The learner trains the ML model and the preliminary accuracy r_s is simultaneously observed by both parties;
3. The seller then charges a price $p_{r_s,t^*} = \max\{LB(r_s, b^*_{r_s}), 0\}$;
4. The machine learner computes his expected utility for the remaining data based on the true prior μ_{ml} and determines whether to accept the offer.

□

F. Derivation of (12)

The expected revenue in (4) is realized as

$$\max_{\mathcal{T}, p_{r_s,\mathcal{T}}} \sum_{r_s} \mathbf{Pr}\left(r_s|\mathcal{T}\right) \cdot \sum_b \varphi(b) \cdot p_{r_s,\mathcal{T}} \cdot \mathbf{1}\Big\{ \sum_q \sum_{r_m} \lambda(r_m|q_{\mathcal{D}}) u(r_m, b) \mathbf{Pr}\left(q|r_s, \mathcal{T}\right) - u(r_s, b) \geq p_{r_s,\mathcal{T}} \Big\}, \tag{23}$$

where $\mathbf{Pr}\left(r_s|\mathcal{T}\right) = \sum_{q_\mathcal{T}} \lambda(r_s|q_\mathcal{T})\mu(q_\mathcal{T})$. The machine learner's estimate for quality can be rewritten as

$$\begin{aligned}
\mathbf{Pr}\left(q|r_s, \mathcal{T}\right) &= \frac{\lambda(r_s|q, \mathcal{T})\mu(q)}{\sum_q \lambda(r_s|q, \mathcal{T})\mu(q)} \\
&= \frac{\lambda(r_s|q, \mathcal{T})\mu(q_\mathcal{R}|q_\mathcal{T})\mu(q_\mathcal{T})}{\sum_{q_\mathcal{T}} \sum_{q_\mathcal{R}} \lambda(r_s|q, \mathcal{T})\mu(q_\mathcal{R}|q_\mathcal{T})\mu(q_\mathcal{T})} \\
&= \mu(q_\mathcal{R}|q_\mathcal{T}) \mathbf{Pr}\left(q_\mathcal{T}|r_s, \mathcal{T}\right),
\end{aligned}$$

where $\mathbf{Pr}\left(q_\mathcal{T}|r_s, \mathcal{T}\right) = \frac{\lambda(r_s|q,\mathcal{T})\mu(q_\mathcal{T})}{\sum_{q_\mathcal{T}} \lambda(r_s|q,\mathcal{T})\mu(q_\mathcal{T})}$. Hence, given the observed accuracy r_s and shared subset $\mathcal{T}$, the remaining expected utility is computed as

$$\sum_q \sum_{r_m} \lambda(r_m|q, \mathcal{D}) u(r_m, b) \mu(q_\mathcal{R}|q_\mathcal{T}) \mathbf{Pr}_s\left(q_\mathcal{T}|r_s, \mathcal{T}\right) - u(r_s, b). \tag{24}$$

Since $\mathcal{T}$ finally should be deterministically chosen by some algorithm, we eliminate subscript $\mathcal{T}$ from $p_{r_s,\mathcal{T}}$ in the rest of discussion for easy exposition. Then (23) equals

$$\begin{aligned}
&\sum_{r_s} \mathbf{Pr}\left(r_s|\mathcal{T}\right) \cdot \sum_b \varphi(b) \cdot p_{r_s} \cdot \mathbf{1}\Big\{ \sum_q \sum_{r_m} \lambda(r_m|q, \mathcal{D}) u(r_m, b) \mathbf{Pr}_s\left(q|r_s, \mathcal{T}\right) - u(r_s, b) \geq p_{r_s} \Big\} \\
=& \sum_{r_s} \mathbf{Pr}\left(r_s|\mathcal{T}\right) \cdot \sum_b \varphi(b) \cdot p_{r_s} \cdot \mathbf{1}\Big\{ \sum_q \sum_{r_m} \lambda(r_m|q, \mathcal{D}) u(r_m, b) \mu(q_\mathcal{R}|q_\mathcal{T}) \mathbf{Pr}_s\left(q_\mathcal{T}|r_s, \mathcal{T}\right) - u(r_s, b) \geq p_{r_s} \Big\} \\
=& \sum_{r_s} \sum_b \varphi(b) \cdot p_{r_s} \cdot 1\Big\{ \sum_q \sum_{r_m} \lambda(r_m|q, \mathcal{D}) u(r_m, b) \mu(q_\mathcal{R}|q_\mathcal{T}) \lambda(r_s|q_\mathcal{T}) \mu(q_\mathcal{T}) - \Big(\sum_{q_\mathcal{T}} \lambda(r_s|q_\mathcal{T}) \mu(q_\mathcal{T}) \Big) u(r_s, b) \geq p_{r_s} \Big\} \\
=& \sum_{r_s} \sum_b \varphi(b) \cdot p_{r_s} \cdot 1\Big\{ \sum_q \sum_{r_m} \lambda(r_m|q, \mathcal{D}) u(r_m, b) \mu(q) \lambda(r_s|q_\mathcal{T}) - \Big(\sum_{q_\mathcal{T}} \lambda(r_s|q_\mathcal{T}) \mu(q_\mathcal{T}) \Big) u(r_s, b) \geq p_{r_s} \Big\},
\end{aligned}$$

where the second equality holds by moving $\mathbf{Pr}\left(r_s|\mathcal{T}\right)$ inside the indicator function and the third equality holds because $\mu(q) = \mu(q_\mathcal{R}|q_\mathcal{T})\mu(q_\mathcal{T})$.

We then focus on the part within the indicator function. Recall that $\lambda(r|q, \mathcal{T})$ is a point distribution defined in (11). Therefore, in the first summation over q, only quality vectors q with $f(q_\mathcal{T}) = r_s$ are left. Additionally, we have $u(r_m, b) = u(f(q), b)$

because $\lambda(r|q,\mathcal{T})$ is a point distribution. Similar analysis is applied to the second term. Then, we have the following formulation

$$\sum_{r_s}\sum_{b}\varphi(b)\cdot p_{r_s}\cdot \mathbf{1}\Big\{\sum_{q|f(q_\mathcal{T})=r_s}\mu(q)u(f(q),b)-\sum_{q_\mathcal{T}|f(q_\mathcal{T})=r_s}\mu(q_\mathcal{T})u(f(q_\mathcal{T}),b)\geq p_{r_s}\Big\}.$$

G. Proof of Theorem 4.1

The proof of Theorem 4.1 mainly consists of Lemma G.1 and Lemma G.4, where Lemma G.1 shows that maximizing (12) subject to a cardinality constraint is NP-hard while Lemma G.4 shows the inapproximation results based on a similar construction as that in Lemma G.1.

G.1. Proof of NP-hardness

Lemma G.1 *It is* NP*-hard to compute the optimal mechanism.*

Proof of Lemma G.1. We observe that maximizing (12) can be equivalently described as follows: The set $\mathcal{D}$ contains $M+1$ features and there are $N+1$ possible choices of quality vectors q. It can be considered as a matrix $\mathcal{D}\in R^{(N+1)\times(M+1)}$. We select a subset $\mathcal{T}$ of features (i.e., columns in matrix $\mathcal{D}$) to maximize the objective function in (12). After that, the N quality vectors are divided into different groups according to $f(q_\mathcal{T})$ of the selected quality subvectors, i.e., given an r_s, quality vectors with $f(q_\mathcal{T})=r_s$ are classified to the same group. The constructed instance $\mathcal{D}$ of size $(N+1)\times(M+1)$ is shown in Figure 3.

The constructed instance $\mathcal{D}$ consists of 1 H-quality vector and N L-quality vectors. As shown in Figure 3, the H-quality vector is of $M+1$ dimensions, which has value H in the last entry and ϵ in other entries. The L-quality vector has value L in the last entry while it has 0 and ϵ in its first M entries. We have the following two assumptions: i) $H>L\gg\epsilon$, and ii) $M\epsilon<L$ and $L+M\epsilon<H$.

We consider a special case of utility function $u(r,b)$ set as follows:

$$u(r,b)=\begin{cases}u(H,b), & r\geq H\\ u(L,b), & L\leq r<H\\ 0, & r<L.\end{cases}\tag{25}$$

Such construction indicates that the last column in D is much more important than others. Therefore, the seller will never share the last dimension with the machine learner, otherwise (12) will give 0 revenue. In other words, if $\mathcal{T}$ includes the $(M+1)^{th}$ column, then the value within the indicator is 0:

$$\begin{aligned}&\sum_{q|f(q_\mathcal{T})=r_s}\mu(q)u(f(q),b)-\sum_{q_\mathcal{T}|f(q_\mathcal{T})=r_s}\mu(q_\mathcal{T})u(f(q_\mathcal{T}),b)\\ =&\sum_{q|f(q_\mathcal{T})=r_s}\mu(q)\Big(u(f(q),b)-u(f(q_\mathcal{T}),b)\Big)\\ =&\ 0.\end{aligned}$$

Therefore, the optimal $\mathcal{T}$ is a subset of the first M features. From our construction in (25), we know $u(f(q_\mathcal{T}),b)=0$ since $f(q_\mathcal{T})<L$. Then, (12) is equivalent to

$$\begin{aligned}\max_{\mathcal{T},\{p_{r_s}\}_{r_s}}\quad&\sum_{r_s}\sum_{b}\varphi(b)\cdot p_{r_s}\cdot 1\Big\{\sum_{q|f(q_\mathcal{T})=r_s}\mu(q)u(f(q),b)\geq p_{r_s}\Big\}\\ \text{s.t.}\quad&|\mathcal{T}|\leq T.\end{aligned}\tag{26}$$

We further assume that $\mu(q)$ is a uniform distribution. Thus, we can safely remove $\mu(q)$ from (26) without affecting our analysis.

By the following lemma, we show that the maximum possible value of the objective function in (26) can be achieved by separating the H-quality vector from all L-quality vectors. "Separate" means that the H-quality vector and the L-quality vectors are divided into different groups according to the value $f(q_\mathcal{T})$.

Lemma G.2 *Suppose there are two quality vectors with sum of quality as H and L, respectively. Recall $u(r,b)$ is increasing in b. Let b_H^* be the optimal private type for $\max_{p_H} \sum_b \varphi(b)\cdot p_H \cdot 1\{u(H,b) \geq p_H\}$ and the solution is $(1-\Phi(b_H^*))u(H,b_H^*)$, as that in Observation A.2. Similarly, b_L^* is optimal type for $\max_{p_L} \sum_b \varphi(b) \cdot p_L \cdot 1\{u(L,b) \geq p_L\}$.*

Assume $b_H^* \neq b_L^*$. We then have higher value by separating H from L, that is

$$\max_{p} \sum_{b} \varphi(b) \cdot p \cdot 1\{u(H,b) + u(L,b) \geq p\} \tag{27a}$$

$$< \quad \max_{p_H} \sum_{b} \varphi(b) \cdot p_H \cdot 1\{u(H,b) \geq p_H\} + \max_{p_L} \sum_{b} \varphi(b) \cdot p_L \cdot 1\{u(L,b) \geq p_L\}. \tag{27b}$$

Proof: First, we show that it is reasonable to assume $b_H^* \neq b_L^*$. We can construct such an example: let i) $b \in \{b_1, b_2\}$ with $b_1 < b_2$ and $\varphi(b_1) = \varphi(b_2) = \frac{1}{2}$, and ii) setting $u(H,b_1) = 10$, $u(H,b_2) = 18$, $u(L,b_1) = 5$ and $u(L,b_2) = 12$. By simple calculation, we know $b_H^* = b_1$ and $b_L^* = b_2$. Thus, the assumption is reasonable.

Second, we know $u(r,b)$ is increasing in b. Thus, there should exist one optimal type b^* at which the (27a) achieves the maximum value $(1-\Phi(b^*))(u(H,b^*)+u(L,b^*))$, where Φ is the cumulative density function of φ.

Similarly, for (27b), we have b_H^* and b_L^* such that (27b) achieves the maximum value, which equals $(1-\Phi(b_H^*))u(H,b_H^*) + (1-\Phi(b_L^*))u(H,b_L^*)$.

Since b_H^* is optimal, we have $(1-\Phi(b^*))u(H,b^*) \leq (1-\Phi(b_H^*))u(H,b_H^*)$. Similar result holds for b_L^*. Combining them, we have

$$\begin{aligned} & \max_{p} \sum_{b} \varphi(b) \cdot p \cdot 1\{u(H,b) + u(L,b) \geq p\} \\ \leq \quad & \max_{p_H} \sum_{b} \varphi(b) \cdot p_H \cdot 1\{u(H,b) \geq p_H\} + \max_{p_L} \sum_{b} \varphi(b) \cdot p_L \cdot 1\{u(L,b) \geq p_L\}. \end{aligned}$$

We can see that the equality holds if and only if $b^* = b_H^* = b_L^*$. Since $b_H^* \neq b_L^*$, we can only have inequality. Thus, we prove the lemma. □

In a similar way, we can prove that for our construction with N L-quality vectors, only when separating N L-quality vectors from the H-quality vector, can we have the maximum value achieved by (26) as

$$\max_{p_H} \sum_{b} \varphi(b) \cdot p_H \cdot 1\{u(H,b) \geq p_H\} + N \cdot \max_{p_L} \sum_{b} \varphi(b) \cdot p_L \cdot 1\{u(L,b) \geq p_L\}. \tag{28}$$

Note that whether separating some L-quality vectors from other L-quality vectors or not does not change the revenue, i.e., $\max_{p_L} \sum_b \varphi(b) \cdot p_L \cdot 1\{N \cdot u(L,b) \geq p_L\} = N \cdot \max_{p_L} \sum_b \varphi(b) \cdot p_L \cdot 1\{u(L,b) \geq p_L\}$.

The NP-hardness is proved by a reduction from the set cover problem (Cormen et al., 2009) defined as: Given a ground set $\mathcal{U} = \{1, 2, \ldots, N\}$, a collection of M subsets of $\mathcal{U}$ and an integer T, determine if there exists a collection $\mathcal{C}$ of at most T subsets so that all the elements in $\mathcal{U}$ are covered (i.e., included) in $\mathcal{C}$?

Given a set cover instance, we construct one instance of (26) as follows: The ground set $\mathcal{U}$ corresponds to the N L-quality vectors, while M subsets correspond to the first M feature columns in D. The given integer T is the cardinality constraint in (26). The feature column is constructed as follows. For the k^{th} feature, given a subset $\mathcal{S} = \{i_1, i_2, \ldots, i_k, \ldots i_s\}$ in the set cover instance, we set the k^{th} entry in the $i_k \in \mathcal{S}$ L-quality vectors as 0 while the k^{th} entry in other L-quality vectors (i.e., $\mathcal{U} \setminus \mathcal{S}$) are set as ϵ. We use one example to explain our construction: Given a set cover instance with a ground set $\mathcal{U} = \{1,2,3\}$, a collection of 3 subsets $\{2\}$, $\{1,3\}$ and $\{1,2\}$. It corresponds to a D of size 4×4, consisting of 1 H-quality vector $[\epsilon, \epsilon, \epsilon, H]$ and 3 L-quality vectors $[\epsilon, 0, 0, L]$, $[0, \epsilon, 0, L]$ and $[\epsilon, 0, \epsilon, L]$.

Note that when selecting subset of feature columns $\mathcal{T}$, the sum of entries in the corresponding quality subvector $f(q_{\mathcal{T}})$ of the H-quality vector is $|\mathcal{T}|\epsilon$, while some of the L-quality vectors have their $f(q_{\mathcal{T}})$ less than $|\mathcal{T}|\epsilon$ because some entries of the selected set of features $\mathcal{T}$ are 0. See Figure 3 for an example: $f(q_{\mathcal{T}}) = |\mathcal{T}|\epsilon = 2\epsilon$ for H-quality vector while that for the first L-quality vector is $f(q_{\mathcal{T}}) = \epsilon$. In this way, we can separate the H-quality vector from L-quality vectors according to the value of $f(q_{\mathcal{T}})$.

Recall that only when separating the H-quality vector from N L-quality vectors (i.e., $\mathcal{U}$ is fully covered), can we achieve the maximum value of (26) as in (28). Therefore, we can answer the decision version of the set cover problem by optimizing (26). Thus, the lemma is proved. □

G.2. Proof of Inapproximation Ratio

The proof of inapproximation ratio makes use of the result from (Feige, 1998), as in the following.

Lemma G.3 *[rephrased Proposition 5.2 in (Feige, 1998)] Given a maximum coverage instance with an integer T as the optimal value (i.e., T is the minimum number of subsets needed so that all the elements in $\mathcal{U}$ are covered) of a set cover instance with the same input, there is no polynomial algorithm giving a coverage of size $(1-\frac{1}{e}+o(1))N$ where N is the number of elements in the ground set.*

Proof: The proof mainly follows from (Feige, 1998). We put it here for completeness. We first show that for such a maximum coverage instance with the given integer T as the optimal value of the corresponding set cover, it is NP-hard to compute the optimal solution. This can be done by contradiction. If computing the optimal solution for this instance is polynomial solvable, then by solving it, we actually can solve the set cover problem in polynomial time. Thus, it is NP-hard.

Assume a polynomial algorithm A approximates this maximum coverage problem within a ratio of $1-1/e+o(1)$. One can show that a polynomial time algorithm B, with A as subroutine, can approximate the set cover problem within $(1-\delta)\ln n$ implying $\mathsf{NP} \subset TIME(n^{O(\log\log n)})$.

Algorithm B repeatedly applies A on the maximum coverage problem, where after each application the points already covered are removed, but T remains unchanged. Since T remains unchanged and is the optimal value of the set cover problem. By the assumption, each time, at least $1-1/e+\epsilon$ of the remaining points are covered. Thus, application of A is at most l times implying $(1/e+\epsilon)^l = 1/n$, where n is the size of the instance. The number of sets used is at most lK. One can obtain that $l = \frac{1}{1-\ln(1-e\epsilon)}\ln n < (1-\delta)\ln n$ for some $\delta > 0$. Then, it means $\mathsf{NP} \subset TIME(n^{O(\log\log n)})$. □

Now, we are ready to prove the inapproximation ratio.

Lemma G.4 *The problem (12) cannot be approximated within a ratio $\frac{e}{e+1}+o(1)$ in polynomial time.*

Proof: We utilize the maximum coverage problem in our proof defined as: Given a ground set $\mathcal{U} = \{1,2,\ldots,N\}$, a collection of M subsets of $\mathcal{U}$ and an integer T, find a collection $\mathcal{C}$ of no more than T subsets such that the number of elements included in $\mathcal{C}$ is maximized. Note that the maximum coverage problem and the set cover problem have the same input.

We construct a maximum coverage instance with integer T as the optimal value of a set cover instance with the same input, as in Lemma G.3. Recall that by Lemma G.3, no polynomial algorithm can reach an approximation ratio $1-\alpha = 1-\frac{1}{e}+o(1)$ for the maximum coverage problem.

The construction of our instance is the same as that in Lemma G.1 except the following change:

- There are only two private types $\{b_1, b_2\}$ for the machine learner and their probabilities are $\varphi(b_1) = 1-\frac{1}{\alpha N}$ and $\varphi(b_2) = \frac{1}{\alpha N}$, respectively. $b_1 < b_2$.
- Let $u(L,b_1) = 1-\frac{1}{N}, u(L,b_2) = 1$, $u(H,b_1) = 1$ and $u(H,b_2) = (\alpha N)^2$.

Recall that the prior over quality vector $\mu(q)$ is uniform distribution. We thus can eliminate $\mu(q)$ without affecting our following analysis. By such construction, we can see that

- for H-vector, $\max_{p_H} \sum_b \varphi(b)\cdot p_H \cdot 1\{u(H,b)\geq p_H\} = (1-\Phi(b_H))u(H,b_H) = \alpha N$, where the optimal type b_H is b_2.
- for L-vectors, $\max_{p_L} \sum_b \varphi(b)\cdot p_L \cdot 1\{u(L,b)\geq p_L\} = 1-\frac{1}{N}$, where the optimal type b_L is b_1.

Thus, the optimal choices of b for the H-quality vector and L-quality vectors satisfy $b_H \neq b_L$. By Lemma G.2, we know that the optimal value is obtained by separating the H-quality vector from L-quality vectors.

We are now ready to show the inapproximability. The optimal solution given constraint $|\mathcal{T}| \leq T$ will be to separate the H-quality vector from N L-quality vectors, which is

$$\begin{aligned}
&\max_{p_H} \sum_b \varphi(b) \cdot p_H \cdot 1\{u(H,b) \geq p_H\} + \max_{p_L} \sum_b \varphi(b) \cdot p_L \cdot 1\{Nu(L,b) \geq p_L\} \\
&= (1-\Phi(b_2))u(H,b_2) + (1-\Phi(b_1))Nu(L,b_1) \\
&= \alpha N + N - 1.
\end{aligned}$$

We know by Lemma G.3 that the polynomial algorithm covers less than $(1-\alpha)N$ elements (i.e., N L-quality vectors), which means that the H-quality vector will be mixed with at least αN L-quality vectors. Suppose that $(1-\alpha')N$ elements are covered where $(1-\alpha) > (1-\alpha')$. The revenue is upper bounded by

$$\begin{aligned}
&\max_{p_{HL}} \sum_b \varphi(b) \cdot p_{HL} \cdot 1\{u(H,b) + \alpha' Nu(L,b) \geq p_{HL}\} \\
&\qquad + \max_{p_L} \sum_b \varphi(b) \cdot p_L \cdot 1\{(1-\alpha')Nu(L,b) \geq p_L\} \\
&< \max_{p_{HL}} \sum_b \varphi(b) \cdot p_{HL} \cdot 1\{u(H,b) + \alpha Nu(L,b) \geq p_{HL}\} \\
&\qquad + \max_{p_L} \sum_b \varphi(b) \cdot p_L \cdot 1\{(1-\alpha)Nu(L,b) \geq p_L\} \\
&= \max_{p_{HL}} \sum_b \varphi(b) \cdot p_{HL} \cdot 1\{u(H,b) + \alpha Nu(L,b) \geq p_{HL}\} + (1-\alpha)(N-1).
\end{aligned}$$

The inequality comes from the fact that mixing the H-quality vector with more L-quality vectors can only decrease the revenue. This is because all L-quality vectors share the same optimal b_L. The proof is similar to that of Lemma G.2.

We will focus on $\max_{p_{HL}} \sum_b \varphi(b) \cdot p_{HL} \cdot 1\{u(H,b) + \alpha Nu(L,b) \geq p_{HL}\}$. If the optimal b is b_1, then it equals

$$\Big(1-\Phi(b_1)\Big)\Big(u(H,b_1) + \alpha Nu(L,b_1)\Big) = 1 + \alpha N - \alpha.$$

If the optimal b is b_2, then it equals

$$\Big(1-\Phi(b_2)\Big)\Big(u(H,b_2) + \alpha Nu(L,b_2)\Big) = \frac{1}{\alpha N}\Big((\alpha N)^2 + \alpha N\Big) = \alpha N + 1.$$

Thus, the maximum possible revenue achieved by a polynomial algorithm is upper bounded by

$$\max_{p_{HL}} \sum_b \varphi(b) \cdot p_{HL} \cdot 1\{u(H,b) + \alpha Nu(L,b) \geq p_{HL}\} + (1-\alpha)(N-1) = N + \alpha.$$

Finally, because $\alpha = \frac{1}{e} - o(1)$, we have ratio for large N

$$\frac{N+\alpha}{\alpha N + N - 1} = \frac{N}{\alpha N + N} + o(1) = \frac{e}{e+1} + o(1). \tag{29}$$

□

H. Omitted Proof in Section 4.2

H.1. Proof of Observation 4.2

Lemma H.1 *In general, the objective function in (12) is not a submodular set function.*

Proof: We prove it by constructing a counter example. Suppose there are 4 vectors: 1 H-quality vector $[\epsilon, \epsilon, \epsilon, \epsilon, H]$ and 3 L-quality vectors $[\epsilon, 0, \epsilon, 0, L], [\epsilon, 0, 0, \epsilon, L]$ and $[\epsilon, \epsilon, 0, 0, L]$. We set uniform prior $\mu(q) = \frac{1}{4}$. Suppose $b \in \{b_1, b_2\}$ with $b_1 < b_2$ and the distribution over $\{b_1, b_2\}$ is uniform, i.e., $\varphi(b) = \frac{1}{2}$. The utility function $u(r, b)$ is defined the same as in (25). Furthermore, we set $u(H, b_1) = 10$, $u(H, b_2) = 18$, $u(L, b_1) = 5$ and $u(L, b_2) = 12$. Hence, the ground set is $\mathcal{U} = \{1, 2, 3, 4, 5\}$.

Denote the revenue in (12) as $F(\mathcal{T})$ where $\mathcal{T}$ is the shared subset of features. Submodularity implies that for any set $S \subseteq T$ and any $l \in \mathcal{U} \setminus T$, $F(S \cup l) - F(S) \geq F(T \cup l) - F(T)$. But in our example, by choosing $S = \{1\} \subseteq T = \{1, 4\}$, we have $F(S) = 6.75$ and $F(T) = 6.75$. By adding $l = \{3\}$ to both sets. $F(S \cup l) = 6.75$ and $F(T \cup l) = 7$. We can see that $F(T \cup l) - F(T) = 0.25 > F(S \cup l) - F(S) = 0$. It contradicts the definition of submodularity. Thus, the objective function in (12) in general is not a submodular function. □

Lemma H.2 *The greedy algorithm (i.e., each step selects one feature that gives the largest increase of revenue) can give arbitrarily bad approximation with respect to the increase of revenue.*

Proof: We show one instance of (12) to which the greedy algorithm gives arbitrarily bad results regarding the increase of revenue. The instance constructed is the same as that in the proof of Lemma H.1. Let $G(S) = F(S) - F(\emptyset)$ be the revenue increase, compared with selling the entire dataset directly according to the prior $\mu(q)$, i.e., $F(\emptyset)$. The seller will not share the 5^{th} feature because $G(\{5\}) = F(\{5\}) - F(\emptyset) < 0$. One optimal solution can be $S_1 = \{3, 4\}$ and we have $G(\{S_1\}) = 0.25$.

However, when applying the greedy algorithm: 1) In the first step, the algorithm computes $G(\{1\}) = G(\{2\}) = G(\{3\}) = G(\{4\}) = 0$ and chooses either one from $\{1, 2, 3, 4\}$; 2) If the algorithm chooses $S_1 = \{1\}$ in the first step, then in the second step, the revenue increase is 0 no matter which feature it chooses, i.e., $G(\{1, 2\}) = G(\{1, 3\}) = G(\{1, 4\}) = 0$. Compared to the optimal revenue increase $G(\{3, 4\}) = 0.25$, the revenue increase $G(\{1, 2\}) = 0$ is arbitrarily bad. □

H.2. Proof of Theorem 4.3

Proof: The revenue obtained by selling the entire dataset without sharing any feature is formulated as

$$\max_{p} \sum_{b} \varphi(b) \cdot p \cdot \mathbf{1}\{\sum_{q} \mu(q) u(f(q), b) \geq p\}. \tag{30}$$

We know that the optimal revenue achieved by (12) is upper bounded by

$$\begin{aligned}
(12) &\leq \max_{\mathcal{T}, \{p_{r_s, \mathcal{T}}\}_{r_s}} \sum_{r_s} \sum_{b} \varphi(b) \cdot p_{r_s, \mathcal{T}} \cdot \mathbf{1}\{\sum_{q | f(q_{\mathcal{T}}) = r_s} \mu(q) u(f(q), b) \geq p_{r_s}\} && \text{(31a)} \\
&\leq \max_{\{p_q\}_q} \sum_{q} \sum_{b} \varphi(b) \cdot p_q \cdot \mathbf{1}\{\mu(q) u(f(q), b) \geq p_q\} && \text{(31b)} \\
&= \sum_{q} \max_{p_q} \sum_{b} \varphi(b) \cdot p_q \cdot \mathbf{1}\{\mu(q) u(f(q), b) \geq p_q\}, && \text{(31c)}
\end{aligned}$$

where (31a) follows by removing the loss of utility term in (12), and (31b) is the possibly maximal value achieved by (31a) through separating all the quality vectors q from each other (Lemma G.2).

To ease the explanation, we consider a special case where $k = 2$, i.e., the learner only has two private types. It can be easily extended to the case $k > 2$. Let b_1, b_2 be two private types where $b_1 < b_2$ and their probabilities are p_1 and p_2 respectively with $p_1 + p_2 = 1$. Recall that $u(\cdot, b)$ is increasing regarding b. By solving the posted price mechanism in (30), we have the optimal value

$$\max\Big\{ \sum_{q} \mu(q) u(f(q), b_1),\ p_2 \sum_{q} \mu(q) u(f(q), b_2) \Big\},$$

and the upper bounded maximum possible revenue i.e., (31c), is

$$\sum_{q} \max \Big\{ \mu(q) u(f(q), b_1),\ p_2 \cdot \mu(q) u(f(q), b_2) \Big\}.$$

We furthermore have

$$\begin{aligned} & \max\Big\{\mu(q)u(f(q),b_1),\ p_2\cdot\mu(q)u(f(q),b_2)\Big\} \\ < \quad & \mu(q)u(f(q),b_1)+p_2\cdot\mu(q)u(f(q),b_2). \end{aligned}$$

Finally, we have

$$\begin{aligned} & \sum_q \max\Big\{\mu(q)u(f(q),b_1),\ p_2\cdot\mu(q)u(f(q),b_2)\Big\} \\ < \quad & \sum_q \mu(q)u(f(q),b_1)+p_2\sum_q \mu(q)u(f(q),b_2) \\ \leq \quad & 2\max\Big\{\sum_q \mu(q)u(f(q),b_1),\ p_2\sum_q \mu(q)u(f(q),b_2)\Big\}. \end{aligned}$$

Therefore, we can see that when $k=2$, the approximation ratio is $\frac{1}{2}$. By similar proof as above, we can extend to the case $k>2$ and obtain approximation ratio $\frac{1}{k}$. □

I. Hardness of Column Subset Selection

The *Column Subset Selection* problem is defined as below:
Input: n arbitrary nonnegative vectors $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n$ of m-dimension.
Output: find a minimum-size set of entries $\mathcal{E} \subseteq [m]$ to distinguish all vectors according to the sum of entries, i.e., $\sum_{i\in\mathcal{E}} \mathbf{x}_i^1 \neq \sum_{i\in\mathcal{E}} \mathbf{x}_i^2 \neq \ldots \neq \sum_{i\in\mathcal{E}} \mathbf{x}_i^n$.

We prove the following theorem,

Theorem I.1 *The* Column Subset Selection *problem is* NP*-hard.*

Proof: As in the proof of binary version of *Column Subset Selection* problem, we reduce from set cover problem with ground set of size n and m sets. Pick n prime numbers $p_1, ..., p_n$ that are all greater than m; note that these numbers can be selected so that they are polynomial in n and m. We create an $(n+1)$-by-m matrix where for $i < n+1$, the entry (i, j) is p_i if the i^{th} element is covered by the j^{th} set and 0 otherwise; the last row consists of all 0s.

Consider a subset of columns, which corresponds to a collection of sets. If element i is uncovered (i.e., not included in any of m sets), the row sum over these selected columns is 0, so row i is indistinguishable from row $n+1$. If i is covered, it is of the form $k \times p_i$, where k is at most m and hence smaller than any of the prime numbers. Thus, the sum in the i^{th} row over these selected columns is divisible by p_i, but not by any p_j with $j \neq i$. Hence, if there selected columns correspond to a cover, all sums are distinct, and otherwise there must exist some row with sum is zero. □

J. Two Examples

The following two examples are modified based on Example 3.3 for justification.

The first example is for the justification of the first order dominance condition. We construct a model distribution such that for any $t_1 \geq t_2$, $\mathbb{E}[r|q,t_1] \leq \mathbb{E}[r|q,t_2]$, i.e., more training data harms the performance. The curves are plotted in Figure 4. We can see that it is optimal for the seller to sell the entire set directly. The revenue of our designed mechanism first drops sharply. Then, it slowly increase, because sharing more data freely gives lower preliminary accuracy and causes less loss of sales value.

Example J.1 *Let $t = 0\%, 1\%, \ldots 100\%$ be the quantity of data. Let $r \in \{0,1,2,\ldots,10\}$ represent $0\%, 10\%, \ldots 100\%$ accuracy, $q \in \{0,1,2,\ldots,10\}$ and private type $b \in \{1,2,\ldots,10\}$. According to the above characterization, let the valuation function be*

$$u(r,b) = \begin{cases} 0, & r+b <= 10 \\ 1000, & r=b=10 \\ 10, & otherwise. \end{cases}$$

The prior belief $\mu(q)$ over q is a Gaussian with standard deviation $\sigma = 3$ and mean $m = 3$. The accuracy distribution $\lambda(r|q,t)$ is also a Gaussian with $m = round(q \times (-t+1))$ and $\sigma = 1.0$. Let $\sigma = 0$ if $q = 0$. Set $\lambda(0|q,0) = 1$.

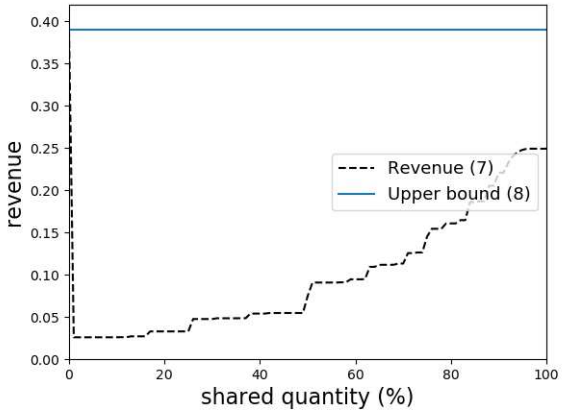

Figure 4. Revenue change with respect to quantity t for Example J.1.

The second example modifies the valuation function to $u(r,b) = [(r+b) \times r]^{0.6}$. The valuation function is almost linear in r but concave in b. By the construction, when r is small, $u(r,b)$ is not negligible. For example, when $b = 10$, comparing $r = 2$ (i.e., 20% accuracy) with $r = 8$, we have $\frac{[(2+8)\times 2]^{0.6}}{[(8+8)\times 8]^{0.6}} \approx 0.3$. It means if the model trained on the shared subset of data achieves 20% accuracy (which in face is a low-accuracy model), the seller may lose 30% of sales value after sharing if the final performance has accuracy 80%, which may lead to less revenue obtained. The revenue curves are plotted in Figure 5, which shows that selling the entire set directly is optimal.

Example J.2 *Let $t = 0\%, 1\%, \ldots 100\%$ be the quantity of data. Let $r \in \{0,1,2,\ldots,10\}$ represent $0\%, 10\%, \ldots 100\%$ accuracy, $q \in \{0,1,2,\ldots,10\}$ and private type $b \in \{1,2,\ldots,10\}$. According to the above characterization, let the valuation function be $u(r,b) = ((r+b) \times r)^{0.6}$. The prior belief $\mu(q)$ over q is a Gaussian with standard deviation $\sigma = 3$ and mean $m = 3$. The accuracy distribution $\lambda(r|q,t)$ is also a Gaussian with $m = round(q \cdot t)$ and $\sigma = 0.1 \times (-(t-0.5)^2 + 0.25)$. Let $\sigma = 0$ if $q = 0$. Set $\lambda(0|q,0) = 1$*

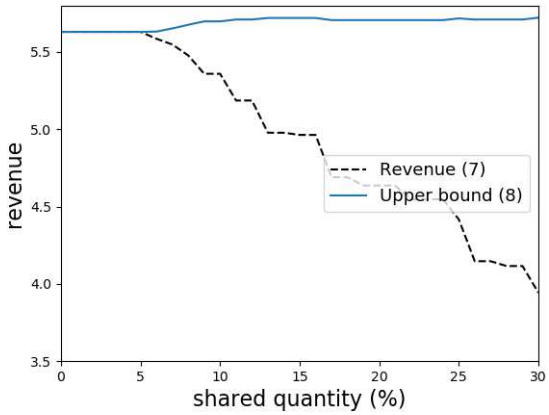

Figure 5. Revenue change with respect to quantity t for Example J.2.

K. Notations Summary

$p_{r_s,\mathcal{T}}$	Price menu. Posted price given shared subset $\mathcal{T}$ of data points and observed preliminary accuracy r_s
q	quality of the data, determined by the model design and data itself. In homogeneous case, it is a scalar. In heterogeneous case, it is a quality vector.
$\mu(q)$	Common prior distribution over q.
$\mu_{sl}(q)$	the seller's private prior belief over q.
$\mu_{ml}(q)$	the machine learner's private prior belief over q.
$u(r, b)$	the machine learner's valuation function of accuracy r and private type b, non-decreasing in r and b.
$\varphi(b)$	Commonly known distribution over private type b.
$\lambda(r\|q, \mathcal{T})$	Probability of outputing accuracy r, conditioning on data quality q and set $\mathcal{T}$.
r, r_s, r_m	r generally denotes the accuracy obtained by ML model; r_s specially denotes the preliminary accuracy observed by the two parties after training on the shared set $\mathcal{T}$; r_m denotes the accuracy obtained by the ML model trained on the whole set $\mathcal{D}$. All of them have the same domain, and are finite and discrete.
$\mathcal{T}$	shared subset of data points
$\mathcal{D}$	universal set of data points
$\mathcal{R}$	complemented set $\mathcal{D} \setminus \mathcal{T}$
$R^{\mathcal{D}}(t)$	possibly maximal revenue obtained from the whole data set $\mathcal{D}$ (i.e., if sharing data cause no loss of sales values), when sharing t quantity of data with the machine learner
$G(r_s, \mathcal{T}, b)$	the seller's expected remaining utility after sharing subset $\mathcal{T}$ of data points and observing accuracy r_s, given the learner's private type b