Locally Differentially Private Personal Data Markets using Contextual Dynamic Pricing Mechanism

Mingyan Xiao, *Student Member, IEEE,* Ming Li, *Member, IEEE,* and Jennifer Jie Zhang

Abstract—Data is becoming the world's most valuable asset and the ultimate renewable resource. This phenomenon has led to online personal data markets where data owners and collectors engage in the data sale and purchase. From the collector's standpoint, a key question is how to set a proper pricing rule that brings profitable tradings. One feasible solution is to set the price slightly above the owner's data cost. Nonetheless, data cost is generally unknown by the collector as being the owner's private information. To bridge this gap, we propose a novel learning algorithm, modified stochastic gradient descent (MSGD) that infers the owner's cost model from her interactions with the collector. To protect owners' data privacy during trading, we employ the framework of local differential privacy (LDP) that allows owners to perturb their genuine data and trading behaviors. The vital challenge is how the collector can derive the accurate cost model from noisy knowledge gathered from owners. For this, MSGD relies on auxiliary parameters to correct biased gradients caused by noise. We formally prove that the proposed MSGD algorithm produces a sublinear regret of $\mathcal{O}(T^{\frac{5}{6}}\sqrt{\log(T^{\frac{1}{3}})})$. The effectiveness of our design is further validated via a series of in-person experiments that involve 30 volunteers.

Index Terms—Local differential privacy, personal data market, private machine learning

1 Introduction

Personal data is recently perceived as a new oil or currency in the digital world. A massive volume of personal data is constantly produced every second. These data are invaluable for public and private sectors to improve their products or services. Demand for personal data for research and business purposes excessively increases while there is practically no efficient supply of these resources. Seeing the commercial opportunities rooted in gaps between demand and supply, the notion of *personal data market* is recently introduced. This notion has transformed perceptions of personal data as undisclosed information to a *commodity*, as noted in [1] and [2]. Several start-up companies, such as Meeco [3], Datacoup [4], and CitizenMe [5], have developed online personal data trading sites and mobile applications following this market orientation.

On one hand, a data agent, also referred as the collector, needs to adequately compensate the data owners for the usage of their personal data or any privacy leakage caused thereby. On the other hand, the collector should properly set the purchasing price to maximize his[1] profit, since overpricing/underpricing can incur loss of profit. So far, most personal data trading platforms employ rather coarse pricing rules. For Datacoup, payment is fixed at approximately $8 per month for accessing seller's social media accounts and financial data (i.e., credit/debit card transactions). This is a similar case for CitizenMe, where owners are paid £0.09 per self-reported demographic, attitudinal and behavioural data. It is questionable whether $8 or £0.09 is reasonable compensation, and how these prices were decided. In view of this issue, we aim to find out proper pricing for the collector for profit maximization in online data markets while compensating owners adequately.

We consider a general personal data market where a data collector interacts with data owners, who arrive in an online fashion, to sell one category of data products, e.g., location trace for the past one week[2]. With a price offered by the collector, the owner compares it with her *data cost*, i.e., the lowest price that she is willing to accept to disclose her data [6]. She tends to accept the offer if the price covers the cost, or reject it otherwise. To model data cost, the concept of *contextual pricing* [7] in computational economics is applied. It states that the cost of a product, personal data here, is a deterministic function of its context features. The data content should be one of the features. For example, people value more for disclosing the information that their car is damaged versus the information that their car is undamaged. Also, the location of the user's home address is more valuable than a public location of a supermarket for the same owner. Another feature counted is the owner's *privacy budget*. A higher budget means greater privacy leakage caused by disclosing the data and thus more cost incurred by selling it to the collector.

As personal data contains sensitive information regarding the owner, directly revealing them would cause irreversible privacy leakage. It is also not rare that a platform may be compromised by adversaries; so is the data stored in the platform. Hence, we propose to apply the *local dif-*

- *M. Xiao is with the Department of Computer Science, California State Polytechnic University, Pomona, CA, 91768.*
- *M. Li is with the Department of Computer Science and Engineering, The University of Texas at Arlington, TX, 76019.*
- *J. Zhang is with College of Business, The University of Texas at Arlington, TX, 76019.*

1. We use "he" to indicate the collector and "she" to indicate the data owner without sex discrimination.

2. We can easily extend our scheme to multiple categories of data products, as each data category is independent.

ferential privacy (LDP) framework to protect owner's data privacy. An owner chooses to perturb her raw data per her privacy protection requirement, before selling it to the collector. Apparently, the utility of the perturbed data would be degraded. Hence, the corresponding payment would be reduced too. In essence, the owner has the full control of how much privacy to sell in data trading. LDP is superior for providing customized privacy protection per owner's perceived privacy budget. Recently, Apple has deployed LDP in macOS Sierra and iOS 10 to gain insight into Apple users' emoji and word usage, while preserving the privacy of individual users [8].

In addition to raw data, an owner's private information also include her data cost and trading behaviors exhibited during transactions. As the data cost is a function of raw data (content), the disclosure of the former provides side information to infer the latter. Regarding the owner's trading behavior, since it is a function of data cost, it also correlates to the data [9]. For example, if an owner turns down an offered price, it indicates that her data cost is even higher, then the data cost range is gradually narrowed down to the true cost. By further given information of the function mapping data cost and its content, owner's trading behavior can possibly reveal partial information regarding the data being traded [10], [11].

To our best knowledge, our paper is the first one that considers the scenario where the collector is untrusted, the data cost/trading behavior is sensitive, and the data collector's desired market property (i.e., profit maximization) and owners' non-negative payoffs[3] are achieved. Nevertheless, prior works just consider part of the above features. Specificaly, one category of prior works [12], [13], [14], [15], [16] assume a trustworthy collector. This category of work achieves the collector's desired market property (e.g., arbitrage-freeness, fairness, profit maximization, or truthfulness) or/and owners' non-negative payoff. They usually compensate data owners in accordance with the data cost [12], [13] or the Shapley value that is the marginal contribution/utility of a data owner [14], [15], [16]. The second category of prior works [17], [18], [19], [20], [21] consider an untrusted collector. They also consider the collector's desired market property or/and owner's non-negative payoff. But none of them think the data cost/trading behavior is sensitive. They propose incentive mechanisms, such as auctions [17], [18], [19], game theory [20], or contracts [21] to elicit data owners to honestly reveal their true cost, as their payment to data owners is a function of data costs.

In our paper, to ensure the collector's profit maximization and owners' non-negative payoff, a feasible strategy is to set the offered price slightly above the owner's data cost– a lower price produces a negative payoff at the owner, while a higher one is less profitable. Although the idea seems to be straightforward, its implementation is faced with a vital challenge–how to predict the data cost on the untrusted collector side, given that data cost is a piece of private information known by the owner herself. Apparently, above incentive mechanisms from the second category [17], [18], [19], [20], [21] are not viable. They will incur extra privacy leakage, as private information, i.e., data costs, are truthfully elicited.

Under the observation that data cost relates to owner's raw data and trading behaviors, i.e., accepting or turning down an offered price, we propose to learn from them the data cost (model). Nonetheless, this is a non-trivial task. As mentioned above, owners can perturb their raw data and trading behaviors using LDP for privacy protection. In other words, knowledge gathered by the collector is noisy. Thus, directly learning over samples with calibrated noise via conventional learning algorithms would lead to inaccurate estimate.

We formulate an online learning problem. All the trading rounds are divided into the exploration phase and the exploitation phase. In the exploration phase, the stochastic gradient descent (SGD) is adopted to approximate owner's cost model. In each round, an owner interacts with the collector her noisy data and trading decision. They are then utilized by the collector to compute the gradient to update the model. As directly applying the noisy samples produces biased gradients, we thus quantify and compensate the bias to reconstruct an unbiased estimate. To facilitate the bias quantification, owners are asked to provide some auxiliary parameters (i.e., the bias and privacy budget) together with their data. We prove that the gap between the derived learning model and the ground truth is bounded within a small margin. The derived model is then leveraged to compute proper pricing for profitable tradings in the exploitation. We modify the existing truncated Laplace based perturbation mechanism. Rather than directly injecting noise to the raw data, noise is first added to gradients. Then an optimization problem is formulated to identify the final perturbed data and auxiliary parameters. We list our key contributions in this paper.

- We study a personal data trading market to maximize the collector's profit while maintaining the owner's non-negative payoffs and tunable privacy. Unlike prior works, we consider the case where the collector is untrustworthy, and the data cost/trading behavior is also sensitive, which renders the problem much more challenging.
- We propose a novel learning algorithm MSGD that enables the collector to learn an accurate model of owner's data cost from their perturbed data and decisions. In particular, the gap between the learning model and the ground truth is bounded to a small margin. As a result, it attains the regret of collector's profit at $\mathcal{O}(T^{\frac{5}{6}}\sqrt{\log(T^{\frac{1}{3}})})$ which is sublinear.
- In addition to the owner's data, the data cost and their trading behaviors are also sensitive and protected under (ϵ, δ)-LDP, which have been neglected in prior works.
- We conduct extensive evaluations of our proposed scheme in terms of learning performance, computation overhead, and communication cost. All evaluations are done based on our dataset that is collected from 30 volunteers over three months.

3. The payoff equals to payment minus cost

2 System Model and Problem Formulation

2.1 System Model

We consider a general system model for personal data trading markets that consists of two kinds of entities: a data collector and a set of data owners. The latter sell personal data to the former in the trade of monetary reward. In real-world applications, a data owner is not necessarily a real person, but an app or a browser add-on that automatically interacts with the collector on behalf of the owner under a pre-configured policy. As the number of collectors does not impact the mechanism design in this work, one collector is considered. Data owners join the market in an online fashion. In each round $t = 1, \cdots, \bar{t}$, the collector trades with one owner. The trading process is described as follows. (1) The collector offers a price p_t. (2) The owner compares p_t with her data cost c_t for revealing her (noisy) data and decides whether to accept or turn down the offer. (3) If the perturbed decision is to accept the offer, the owner submits her noisy data and receives p_t from the collector accordingly; otherwise, the trading is aborted. An owner is allowed to participate in multiple rounds of trading. In a new round, the owner is required to provide the data that has not been sold before, either a newly generated one or the one from a previously aborted trading.

In this paper we adopt the *contextual pricing* in computational economics to model owner's data cost c_t. It states that the cost of a product, personal data here, is a deterministic function of its contextual features. Let $\boldsymbol{d}_t$ denote the owner's n-dimensional raw binary data vector, e.g., the location trace to be traded on t-iteration. $(\epsilon, \delta) \in [0, 1]^2$ be her privacy budget. We also discuss in the supplemental file how to extend our design to data in numerical and categorical formats. Then $\boldsymbol{d}_t$, ϵ, and δ are deemed the contextual features. In specific, (ϵ, δ) decides how much noise to inject to the raw data. Intuitively, the more "noisy" the reported data is, the more privacy is retained, and thus the less cost is incurred. Denote by $\boldsymbol{x}_t = [\boldsymbol{d}_t, \epsilon, \delta]^\top$ the owner's contextual feature set. Following prior works on data trading [19], [22], data cost is formulated as the linear function of $\boldsymbol{x}_t$, i.e., $c_t = \boldsymbol{w}^{*\top} \boldsymbol{x}_t$. The discussion is further extended to nonlinear model in the supplemental file. $\boldsymbol{x}_t$ is assumed to draw according to a fixed and underlying distribution D, with $\boldsymbol{w}^* \in \mathcal{W} = \{\boldsymbol{w} : \|\boldsymbol{w}\|_2 \leq \frac{1}{\sqrt{n+2}}\}$ and $\boldsymbol{w}^{*\top} \boldsymbol{x}_t \leq 1$ [4].

With the emergence of data trading, how to allow individuals to estimate the cost of revealing a piece of personal data has been investigated extensively in experimental economics (e.g., [6], [23], [24]). With this basis, a data owner is deemed to be aware of her data cost in this work.

2.2 Preliminaries

Local differential privacy (LDP). Given a a pair of privacy parameter (ϵ, δ) that controls the privacy disclosure, a randomized function $\mathcal{A}$ satisfies (ϵ, δ)-LDP, defined as follows.

4. In fact, we can have a generalized form $\mathcal{W} = \{\boldsymbol{w} : \|\boldsymbol{w}\|_2 \leq W\}$ and $\boldsymbol{w}^{*\top} \boldsymbol{x}_t \leq W \|\boldsymbol{x}_t\|_2 \leq W\sqrt{n+2}$. The upper bound of c_t can still be transformed to 1 by rescaling $\boldsymbol{w}$ via $\boldsymbol{w}/(W\sqrt{n+2})$.

TABLE 1
Notations.

$\boldsymbol{d}_t$	raw data	$\tilde{\boldsymbol{d}}_t$	perturbed data
t	trading index	ξ	extra payment
$\boldsymbol{g}_t^*$	optimum gradient	$\boldsymbol{g}_t$	biased gradient
$\boldsymbol{g}_t'$	corrected gradient	$\tilde{\boldsymbol{g}}_t$	noisy gradient
ϵ, δ	privacy parameters	c_t	data cost
$\boldsymbol{l}_t, \boldsymbol{r}_t$	auxiliary parameters	p_t	offered price
a_t	perturbed decision	$\tilde{\boldsymbol{x}}_t$	$[\boldsymbol{d}_t, \epsilon, \delta]$
α	fraction of exploration stage	$\boldsymbol{x}_t$	$[\boldsymbol{d}_t, \epsilon, \delta]$
$\boldsymbol{w}^*$	ground truth model	$\boldsymbol{w}_t$	estimate of $\boldsymbol{w}^*$
τ	no. of data traded in explor.	$\mathbf{1}_{\{p_t \geq c_t\}}$	raw decision
T	no. of data traded over $\bar{t}$	$\bar{t}$	total no. of trad.

Definition 1. ((ϵ, δ)-LDP). *A randomized function $\mathcal{A}$ satisfies (ϵ, δ)-LDP iff for any two arbitrary inputs $(\boldsymbol{x}, \boldsymbol{x}')$ and for any possible output $\mathcal{S}$ of $\mathcal{A}$, we have*

$$\Pr[\mathcal{A}(\boldsymbol{x}) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{A}(\boldsymbol{x}') \in \mathcal{S}] + \delta.$$

LDP is a special case of differential privacy [25] where the random perturbation is performed by the data owner, not by the collector. In other words, the collector never possesses the exact private data of any data owner. According to the above definition, the collector, who receives the perturbed data, cannot distinguish whether the true input is $\boldsymbol{x}$ or $\boldsymbol{x}'$ with high confidence (controlled by the parameter ϵ and δ). Here, ϵ and δ are called *privacy budget* that control the strength of privacy protection, depending on owner's respective privacy requirements. A smaller ϵ indicates strong privacy protection because the adversary has lower confidence when trying to distinguish any pair of inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$. δ is a small probability that allows the upper bound that ϵ does not hold. Hence, a smaller δ indicates a stringent privacy requirement. When δ becomes 0, (ϵ, δ)-LDP is transformed to conventional ϵ-LDP. Two well-known composition properties of LDP will be used in this paper, *sequential composition* and *parallel composition*.

Theorem 1. (Sequential composition [25]). *If a randomized mechanism $\mathcal{M}_i$ satisfies (ϵ_i, δ_i)-LDP for $i \in [1, k]$, then their sequential composition $\mathcal{M}(D) = (\mathcal{M}_1(D), \cdots \mathcal{M}_k(D))$ satisfies $(\sum_i \epsilon_i, \sum_i \delta_i)$-LDP, given the input dataset D.*

According to sequential composition, given a privacy budget, a computation task can be split into multiple portions, where each portion corresponds to the budget for a sub-task.

Theorem 2. (Parallel composition [26]). *A randomized mechanism $\mathcal{M}_i$ satisfies (ϵ_i, δ_i)-LDP. Let $D_i, \forall i \in [1, k]$ be arbitrary disjoint subsets of the input domain D. The parallel composition $\mathcal{M}(D) = (\mathcal{M}_1(D_1), \cdots, \mathcal{M}_k(D_k))$ provides $(\max_i(\epsilon_i), \max_i(\delta_i))$-LDP.*

Parallel composition provides privacy bound of multiple parallel mechanisms. Specifically, if the domain of input data is partitioned into disjoint sets which are subjected to differential private analysis, the ultimate privacy guarantee depends only on the worst guarantee of each analysis.

2.3 Problem Statement

Problem statement at data owners. The goal of a data owner is to protect her privacy from adversaries including the collector during tradings. Besides personal data $\boldsymbol{d}_t$, owner's raw trading behaviors also bear sensitive information regarding the data. To be specific, an owner's raw decision is to accept an offer p_t if $p_t \geq c_t$; reject the offer p_t otherwise. Recall that c_t is a function of $\boldsymbol{x}_t$ ($\boldsymbol{d}_t$). Thus, collector can compromise data privacy through analyzing owner's trading behaviors. The framework of (ϵ, δ)-LDP is applied. Each owner perturbs her released information by injecting calibrated noise according to her privacy budget (ϵ, δ).

Problem statement at data collector. Let v_t be the collector's perceived value for purchasing data in each round t. Then, the collector's profit, denoted by r_t, is calculated by $r_t = a_t(v_t - p_t)$, i.e., the difference between v_t and the incurred payment p_t. a_t is a binary (noisy) trading decision ($a_t = 1$ if the owner accepts the offer, and 0 otherwise). The collector's goal is to select proper payment p_t in each round t that maximizes his overall (expected) profit, equivalently minimizes the (expected) regret $R(T) = \sum_{t=1}^{\bar{t}}(r_t^* - \mathbb{E}[r_t])$, where $r_t^* = \max \mathbb{E}[r_t]$ [27]. The expectation is with respect to owner's random data. Denote by T ($T > \bar{t}$) the total number of data records the collector purchased during $\bar{t}$ trading rounds. $R(T)$ can be interpreted as the difference between the collector's maximum expected profit and the actual one (by offering p_t) over T total data records. The definition of $R(T)$ here is slightly different from the conventional one where T denotes the total iteration rounds. Owners are allowed to trade multiple data records in one round.

Proposition 1. *The collector's expected regret,* $R(T) = \sum_{t=1}^{\bar{t}}(r_t^* - \mathbb{E}[r_t])$*, is minimized at* $p_t = c_t$ ($t \in [1, \cdots, \bar{t}]$).

All proofs are given in the supplemental file. As stated in Proposition 1, the minimum regret is attained when the collector sets the price p_t the same as owner's data cost c_t. Nonetheless, c_t is unknown to the collector as it is private information to the owner for being a function of the raw data $\boldsymbol{d}_t$ ($\boldsymbol{x}_t$). To circumvent this, the collector learns the proper price p_t from owner's noisy data $\tilde{\boldsymbol{d}}_t$ ($\tilde{\boldsymbol{x}}_t$) and perturbed trading behaviors a_t. It would be desirable if p_t leads to sublinear regret. i.e., the averaged $R(T)$ for each trading data record diminishes and approximates to 0 as $T \to \infty$.

Design overview at data collector. Our scheme starts from learning $\boldsymbol{w}^*$ via minimizing a *surrogate loss function* for which $\boldsymbol{w}^*$ is the minimizer. Given $c_t = \boldsymbol{w}^{*\top}\boldsymbol{x}_t$, once $\boldsymbol{w}^*$ is derived, so is c_t. The basic framework of Stochastic Gradient Descent (SGD) [28] is applied. A salient difference from existing online learning scenarios is that samples are perturbed with owner's calibrated noise here. As a result, the conventional SGD online learning process is biased. To resolve this issue, we first quantify the bias caused by noisy data. Then, in the exploration phase, the gradient is adapted to compensate for the bias. The learning model is updated through the unbiased gradient to approximate $\boldsymbol{w}^*$ gradually. In the exploitation phase, the collector utilizes the pricing rule, calculated from the learning result over $\boldsymbol{w}^*$, to purchase data.

Design overview at data owners. We intend to release the collector's required information to cooperate with the collector and facilitate trading, while achieving (ϵ, δ)-LDP in both the exploration phase via $\mathcal{A}_r$ and the exploitation phase via $\mathcal{A}_t$. In the exploration phase, as mentioned above, the learning algorithm at the collector needs to be aware of the bias of gradients caused by noisy samples. For this purpose, a data owner first injects calibrated noise to the optimal rate of gradient descent. Then, the gap between the optimal rate and the one derived from perturbed data is quantified. She then resorts to an optimization problem that minimizes the gap. The optimum results include the final perturbed data and nonsensitive auxiliary parameters (e.g., the gradient bias), which are all forwarded to the collector. The proposed data perturbation method is different from the conventional one which directly injects noise to the raw data. In the latter, gradients are biased. Since the bias contains information regarding the raw data, it cannot be released. Also, existing SGD based differentially private learning algorithms where noisy gradients are submitted to the collector are not viable here, as the commodity in real-world personal data markets is the data record itself rather than query results (e.g., gradients). Details will be discussed in Section 4.1. The randomization schemes for trading decision perturbation in both the exploration and exploitation phases and data perturbation in the exploitation phase mainly follow *random response* (RR) to attain LDP.

Structure of following sections. The entire design consists of the collector-side learning algorithm and the owner-side perturbation schemes. The former is elaborated in Section 3. Since the perturbation schemes for the exploration phase and the exploitation phase are distinct, we present them in Section 4 and Section 5, respectively. Theoretical analysis is provided in Section 6. Table 1 summarizes the notions that are frequently mentioned.

3 Online Learning Algorithm for Collector

According to Proposition 1, the collector's goal is to infer $\boldsymbol{w}^*$ and thus c_t. Then $R(T)$ is minimized by setting p_t equal to c_t in each round t. Since $R(T)$ is not convex, differentiable, or even continuous, it is hard to work on it directly. We thus resort to a *surrogate loss function* $F(\boldsymbol{w}_t) = \mathbb{E}_{\boldsymbol{x}_t \sim D}[(\boldsymbol{w}_t^\top \boldsymbol{x}_t - \boldsymbol{w}^{*\top}\boldsymbol{x}_t)^2]$, whose minimizer is exactly $\boldsymbol{w}^*$. The collector minimizes F so as to infer $\boldsymbol{w}^*$ in an online learning fashion.

We now introduce the notion of *strong convexity*. A twice-differentiable function $H(\boldsymbol{w})$ is λ-strongly convex if and only if the Hessian matrix $\nabla^2 H(\boldsymbol{w})$ is full rank and the minimum eigenvalue of $\nabla^2 H(\boldsymbol{w})$ is at least λ. Note that F is strongly convex if and only if the covariance matrix of the data is full-rank, since $\nabla^2 F(\boldsymbol{w}_t) = 2\mathbb{E}_{\boldsymbol{x}_t \sim D}[\boldsymbol{x}_t \boldsymbol{x}_t^\top]$. We make the following assumption throughout the paper.

Assumption 1. *The minimum eigenvalue of* $2\mathbb{E}_{x_t \sim D}[\boldsymbol{x}_t \boldsymbol{x}_t^\top]$ *is at least* λ.

If the above assumption does not hold, then there is redundancy in the attributes and the data can be projected into a lower dimensional space with a full-rank covariance matrix (for example using PCA) and without any loss in information [29].

Algorithm 1 A naive algorithm

1: **Input:** $0 \leq \alpha \leq 1$, $\boldsymbol{w}_1 = \mathbf{1} \in \mathcal{W}$, $T_\alpha = \lceil \alpha T \rceil$;
// Exploration phase
2: **for** $t = 1, \cdots, T_\alpha$ **do**
3: Publish the pricing rule: $p_t = \boldsymbol{w}_t^\top \tilde{\boldsymbol{x}}_t$;
4: Observe a_t;
5: **if** $a_t = 1$ **then**
6: Collect $\tilde{\boldsymbol{x}}_t$;
7: $\mathbf{g}_t = 2(\boldsymbol{w}_t^\top \tilde{\boldsymbol{x}}_t - c_t)\tilde{\boldsymbol{x}}_t$;
8: $\boldsymbol{w}_{t+1} = \prod_{\mathcal{W}}(\boldsymbol{w}_t - \eta_t \mathbf{g}_t)$, where $\eta_t = 1/\lambda t$;
9: **else**
10: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t$;
11: **end if**
12: **end for**
// Exploitation phase
13: **for** $t = T_\alpha + 1, \cdots, T$ **do**
14: Publish the pricing rule: $p_t = \boldsymbol{w}_{T_\alpha+1}^\top \tilde{\boldsymbol{x}}_t$;
15: **if** $a_t = 1$ **then**
16: Collect $\tilde{\boldsymbol{x}}_t$;
17: **end if**
18: **end for**

Algorithm 2 The MSGD Algorithm

1: **Input:** $0 \leq \alpha \leq 1$, $\boldsymbol{w}_1 = \mathbf{0} \in \mathcal{W}$, $T_\alpha = \lceil \alpha T \rceil$, $\tau = 0$;
// Exploration phase
2: **for** $t = 1, \cdots, T_\alpha$ **do**
3: Offer $p_t \sim U[0, 2]$;
4: Observe a_t;
5: **if** $a_t = 1$ **then**
6: $\tau = \tau + 1$;
7: Collect $\tilde{\boldsymbol{x}}_t$, $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$;
8: $\boldsymbol{g}'_t = \boldsymbol{g}_t - \boldsymbol{r}_t + \boldsymbol{l}_t = 2(\boldsymbol{w}_t^\top \tilde{\boldsymbol{x}}_t)\tilde{\boldsymbol{x}}_t - \boldsymbol{r}_t + \boldsymbol{l}_t$;
9: $\boldsymbol{w}_{t+1} = \prod_{\mathcal{W}}(\boldsymbol{w}_t - \eta_t \boldsymbol{g}'_t)$, where $\eta_t = \frac{1}{\lambda \tau}$;
10: **else**
11: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t$;
12: **end if**
13: **end for**
// Exploitation phase
14: **for** $t = T_\alpha + 1, \cdots, \bar{t}$ **do**
15: Publish pricing rule $p_t = p_t(\tilde{\boldsymbol{d}}_t, \boldsymbol{w}_{T_\alpha+1}, \xi)$;
16: Observe a_t;
17: **if** $a_t = 1$ **then**
18: Collect $\tilde{\boldsymbol{x}}_t$;
19: **end if**
20: **end for**

3.1 A Naive Online Learning Approach

We start from a naive approach that employs the well-known stochastic gradient descent (SGD) online learning algorithm [28]. It conducts a stochastic gradient descent to minimize the surrogate loss function F.

The naive algorithm consists of exploration and exploitation phases. A parameter $\alpha \in [0, 1]$ determines the fraction of transactions that are spent in the exploration phase as oppose to the exploitation phase. Like many online learning algorithms, a large α indicates more learning rounds, and the opposite at a small α. In the exploration phase, the goal is to minimize F over some convex domain $\mathcal{W}$. The collector utilizes a stochastic gradient oracle, which given some $\boldsymbol{w}_t \in \mathcal{W}$, produces a vector $\mathbf{g}_t = 2(\boldsymbol{w}_t^\top \tilde{\boldsymbol{x}}_t - c_t)\tilde{\boldsymbol{x}}_t$. $\boldsymbol{w}_{t+1}$ is updated via $\prod_{\mathcal{W}}(\boldsymbol{w}_t - \eta_t \mathbf{g}_t)$. η_t is the learning rate that is set to $1/\lambda t$. $\prod_{\mathcal{W}}$ is the projection operator on $\mathcal{W}$. Note that $\boldsymbol{w}_{t+1}$ is only updated when a new data record is collected from the data owner, i.e., the owner accepts the offered price. In the exploitation phase, the pricing rule is published as $p_t = \boldsymbol{w}_{T_\alpha+1}^\top \tilde{\boldsymbol{x}}_t$, with $\boldsymbol{w}_{T_\alpha+1}$ the learned value toward $\boldsymbol{w}^*$ from the last round of the exploration phase. The owner computes the payment p_t, compares it with the data cost c_t, and decides whether to sell her data. α is the learning ratio that strikes the exploration-exploitation trade-off.

There is a limitation in the naive algorithm. The SGD online learning algorithm requires $\mathbf{g}_t$ as an unbiased estimate of $\nabla F(\boldsymbol{w}_t)$, i.e., $\mathbb{E}[\mathbf{g}_t|\boldsymbol{w}] = \nabla F(\boldsymbol{w}_t)$ with $\nabla F(\boldsymbol{w}_t) = \frac{\partial F}{\partial \boldsymbol{x}_t} = \mathbb{E}_{\boldsymbol{x}_t \sim D}[2(\boldsymbol{w}_t^\top \boldsymbol{x}_t - c_t)\boldsymbol{x}_t]$. Otherwise, $\boldsymbol{w}_{T_\alpha+1}$ cannot be a good approximation of $\boldsymbol{w}^*$. This is not a concern if the collected samples are clean. In our scenario, the collector only has access to perturbed data $\tilde{\boldsymbol{x}}_t$. Hence, $\mathbf{g}_t$, calculated based on $\tilde{\boldsymbol{x}}_t$, cannot have the above condition hold. Now that $\boldsymbol{w}_{T_\alpha+1}$ is a erroneous estimate of $\boldsymbol{w}^*$, the offered price derived accordingly (line 14) can hardly produce the regret at $o(T)$.

3.2 Our Online Learning Algorithm

The key contribution of our algorithm lies in quantifying and remedying the bias in gradients caused by perturbed data. First of all, we need to identify one unbiased estimate of $\nabla F(\boldsymbol{w}_t)$. Let $\boldsymbol{g}_t^* = 2(\boldsymbol{w}_t^\top \boldsymbol{x}_t - 2(1 - \mathbf{1}_{\{p_t \geq c_t\}}))\boldsymbol{x}_t$, where $p_t \sim U[0, 2]$ and $\mathbf{1}_A$ denotes the indicator function of some event A. $\boldsymbol{g}_t^*$ is an unbiased estimate of the gradient of F, as stated by the following proposition.

Proposition 2. *The random variable $\boldsymbol{g}_t^*$ satisfies $\mathbb{E}[\boldsymbol{g}_t^*|\boldsymbol{w}_t] = \nabla F(\boldsymbol{w}_t)$.*

According to [28], $\boldsymbol{g}_t^*$ can update $\boldsymbol{w}_t$ that converges to $\boldsymbol{w}^*$ with sufficient iterations. Note that $\boldsymbol{g}_t^*$ is computed from owner's raw data $\boldsymbol{x}_t$ and decision $\mathbf{1}_{\{p_t \geq c_t\}}$. As the collector is only aware of the perturbed data $\tilde{\boldsymbol{x}}_t$ and decision a_t, $\boldsymbol{g}_t^*$ cannot be directly obtained. Instead, the collector can compute $\boldsymbol{g}_t = 2(\boldsymbol{w}_t^\top \tilde{\boldsymbol{x}}_t - 2(1 - a_t))\tilde{\boldsymbol{x}}_t$ (by replacing $\boldsymbol{x}_t$ and $\mathbf{1}_{\{p_t \geq c_t\}}$ with $\tilde{\boldsymbol{x}}_t$ and a_t in $\boldsymbol{g}_t^*$) [5]. As $\mathbb{E}[\boldsymbol{g}_t|\boldsymbol{w}_t] \neq \nabla F(\boldsymbol{w}_t)$, $\boldsymbol{g}_t$ is biased. Luckily, we are able to quantify the bias $\mathbb{E}[\boldsymbol{g}_t|\boldsymbol{w}_t] - \nabla F(\boldsymbol{w}_t) = \boldsymbol{r}_t - \boldsymbol{l}_t$, where $\boldsymbol{r}_t$ and $\boldsymbol{l}_t$ are two auxiliary vectors submitted by the data owner. As $\mathbb{E}[\boldsymbol{g}_t - \boldsymbol{r}_t + \boldsymbol{l}_t|\boldsymbol{w}_t] = \nabla F(\boldsymbol{w}_t)$ (proved in Lemma 3), then $\boldsymbol{g}_t - \boldsymbol{r}_t + \boldsymbol{l}_t$, denoted by $\boldsymbol{g}'_t$, can serve as an unbiased estimate of gradient of F. Then, being updated by $\boldsymbol{g}'_t$, $\boldsymbol{w}_t$ can approximate $\boldsymbol{w}^*$. The values of $\boldsymbol{r}_t$ and $\boldsymbol{l}_t$ are dependent on perturbation schemes adopted by the data owner. Their instantiation will be discussed in Section 4.1.

We are now ready to present our algorithm which is called modified SGD (MSGD). It follows the basic framework of SGD online learning [28], but differs in the following key aspects. (1) In the exploration phase, the offered prices follow uniform distribution (line 3). The rationale behind is to allow the collector to infer owner's cost model via their responses toward various offers. More specific, the collector intentionally offers uniformly distributed prices p_t so that, in each round, the user's behavior reveals the gradient of the surrogate loss at our current estimate for $\boldsymbol{w}^*$. (2) As discussed, since $\boldsymbol{g}'_t$ is an unbiased gradient of F, MSGD uses it to update $\boldsymbol{w}_t$ that approximates $\boldsymbol{w}^*$. (3) In the exploitation phase, the pricing rule is published as $p_t = p_t(\tilde{\boldsymbol{d}}_t, \boldsymbol{w}_{T_\alpha+1}, \xi)$ (line 15). Its explicit expression is given in (1). ξ is a small value that ensures sublinear regret of MSGD. Details are provided in Section 5.

Bounding learning gap. Now that $\boldsymbol{g}'_t$ is an unbiased gradient, we readily have the following result that bounds the gap between $\boldsymbol{w}_t$ and $\boldsymbol{w}^*$.

Lemma 1. (Revised from Proposition 1 of [28]). *Let $\sigma \in (0, 1/e)$ and $\tau > 4$. Suppose F is λ-strongly convex over a convex set $\mathcal{W}$, and that $\|\boldsymbol{g}'_t\|_2^2 \leq G^2$. Then it holds that $\|\boldsymbol{w}_\tau - \boldsymbol{w}^*\|_2^2 \leq \frac{(624 \log(\log(\tau)/\sigma) + 1)G^2}{\lambda^2 \tau}$ with a probability at least $1 - \sigma$. Note that τ is the total number of data records successfully traded during the exploration phase.*

5. $a_t = \mathbf{1}_{\{p_t \geq c_t\}}$ if the trading decision is not perturbed by the owner.

Lemma 1 guarantees that, with high probability, the distance between the learned parameter vector $\boldsymbol{w}_t$ and the target $\boldsymbol{w}^*$ is bounded. G is negatively correlated with ϵ and δ. Thus, a higher privacy budget, i.e., larger (ϵ, δ) and thus less noisy data, leads to a smaller learning gap. The exact value of G^2 is presented in Proposition 4 of the appendix. Note that the ℓ_2 distance bound in Lemma 1 serves as the cornerstone of deriving ξ and thus the price p_t. (Details are provided in Section 5).

Again, the training samples (data) in SGD online learning [28] is noiseless. Based on them, it is straightforward to compute the unbiased gradient of F. In our case, the collector has access to perturbed data only. Nevertheless, as one of the contributions of our MSGD, it can still derive unbiased gradient $\boldsymbol{g}'_t$ to update $\boldsymbol{w}_t$ so as to approximate $\boldsymbol{w}^*$. That is why we have Lemma 1 via minor modification of Proposition 1 in [28].

4 Perturbation in Exploration Phase

The randomized algorithm in the exploration phase $\mathcal{A}_r$ consists of two stages. In stage I (II), it perturbs owner's raw data $\boldsymbol{d}_t$ (trading decision $\mathbf{1}_{\{p_t \geq c_t\}}$) into noisy one $\tilde{\boldsymbol{d}}_t$ (a_t). We then denote by $\mathcal{A}_{r,I}$ and $\mathcal{A}_{r,II}$ the randomized algorithms developed for these two stages, respectively. In specific, we aim to attain (ϵ_I, δ)-LDP for $\mathcal{A}_{r,I}$ and ϵ_{II}-LDP for $\mathcal{A}_{r,II}$, where $\epsilon_I + \epsilon_{II} = \epsilon$.

4.1 Stage I: Data Perturbation

Noise is first injected to the optimal gradient of F. We then resort to an optimization problem with perturbed gradient as its inputs. The optimum solution includes the final perturbed data. Here we give the formal definition of the data perturbation algorithm $\mathcal{A}_{r,I}$.

Definition 2. (Randomized algorithm $\mathcal{A}_{r,I}$). *Given the raw data $\boldsymbol{d}_t$ as the input, the randomized algorithm $\mathcal{A}_{r,I}$ outputs the perturbed data $\tilde{\boldsymbol{d}}_t$ and auxiliary parameters $\boldsymbol{l}_t$, $\boldsymbol{r}_t$.*

Generating noised stochastic gradients. Recall that $\boldsymbol{g}_t^*$ is an unbiased estimate of ∇F. It is calculated as $\boldsymbol{g}_t^* = 2(\boldsymbol{w}_t^\top \boldsymbol{x}_t - 2(1 - \mathbf{1}_{\{p_t \geq c_t\}}))\boldsymbol{x}_t$. Inspired by the one-dimension truncated Laplacian mechanism [30], a $(n+2)$-dimension noise $\boldsymbol{Z}$ is generated, with each element following a truncated Laplacian distribution:

$$f(Z|\Delta_1 \boldsymbol{g}_t^*, \epsilon_I, \delta) = \begin{cases} \frac{1}{2\frac{\Delta_1 \boldsymbol{g}_t^*}{\epsilon_I}(1 - \frac{1}{1+\frac{e^{\epsilon_I}-1}{2\delta}})} e^{-|Z|/(\Delta_1 \boldsymbol{g}_t^*/\epsilon_I)}, & Z \in [-A, A], \\ 0, & \text{otherwise}, \end{cases}$$

where $A = \frac{\Delta_1 \boldsymbol{g}_t^*}{\epsilon_I} \log(1 + \frac{e^{\epsilon_I}-1}{2\delta})$ and $\Delta_1 \boldsymbol{g}_t^*$ is the ℓ_1 sensitivity of $\boldsymbol{g}_t^*$. Then the data owner first generates noisy gradient vector $\tilde{\boldsymbol{g}}_t$ as $\tilde{\boldsymbol{g}}_t = \boldsymbol{g}_t^* + \boldsymbol{Z} = 2(\boldsymbol{w}_t^\top \boldsymbol{x}_t - 2(1 - \mathbf{1}_{\{p_t \geq c_t\}}))\boldsymbol{x}_t + \boldsymbol{Z}$. Here we choose truncated Laplace noise over Laplace noise adopted in conventional differential privacy mechanism. This is because the former bears bounded amplitude, while the latter does not. Noise with bounded attitude produces bounded gradients in MSGD which is essential to bound $\|\boldsymbol{w}_\tau - \boldsymbol{w}^*\|_2^2$ in Lemma 1. $\Delta_1 \boldsymbol{g}_t^*$ is defined as $\max \|\boldsymbol{g}_t^*(\boldsymbol{d}_t) - \boldsymbol{g}_t^*(\boldsymbol{d}'_t)\|_1$, where $\boldsymbol{d}_t$ and $\boldsymbol{d}'_t$ are two arbitrary data records from the data owner. The following proposition quantifies $\Delta_1 \boldsymbol{g}_t^*$.

Proposition 3. *The ℓ_1 sensitivity of $\boldsymbol{g}_t^*$ ($\Delta_1 \boldsymbol{g}_t^*$) is $8(n+2)$.*

Generating the output of $\mathcal{A}_{r,I}$. The following part takes the above noisy gradient as input and generates the perturbed data $\boldsymbol{d}_t$ and auxiliary parameters $\boldsymbol{l}_t$, $\boldsymbol{r}_t$. As $\mathbb{E}[\tilde{\boldsymbol{g}}_t|\boldsymbol{w}_t] = \mathbb{E}[\boldsymbol{g}_t^*|\boldsymbol{w}_t] = \nabla F(\boldsymbol{w}_t)$, $\tilde{\boldsymbol{g}}_t$ is an unbiased estimate of $\nabla F(\boldsymbol{w}_t)$. Following the result of [28], the online SGD converges $\boldsymbol{w}^*$ to with $\boldsymbol{w}_t$ updated in a way $\boldsymbol{w}_{t+1} = \prod_{\mathcal{W}}(\boldsymbol{w}_t - \eta \tilde{\boldsymbol{g}}_t)$ given sufficient iterations. On the other hand, as the data collector is only aware of perturbed data $\tilde{\boldsymbol{d}}_t$ and decision a_t, $\tilde{\boldsymbol{g}}_t$ is not accessible but $\boldsymbol{g}_t$. Recall that $\boldsymbol{g}_t = 2(\boldsymbol{w}_t^\top \tilde{\boldsymbol{x}}_t - 2(1 - a_t))\tilde{\boldsymbol{x}}_t$. It is degraded to $2(\boldsymbol{w}_t^\top \tilde{\boldsymbol{x}}_t)\tilde{\boldsymbol{x}}_t$ when the owner accepts the offer, i.e., $a_t = 1$. As discussed in Section 3.2, $\boldsymbol{g}_t$ is a biased estimate of $\nabla F(\boldsymbol{w}_t)$. To remedy the bias, we first quantify the difference between $\tilde{\boldsymbol{g}}_t$ and $\boldsymbol{g}_t$. Let $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$ be two non-negative vectors. The difference is expressed as $-\boldsymbol{l}_t \leq \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t \leq \boldsymbol{r}_t$. We then formulate the following optimization problem, denoted as P_1

$$\begin{aligned} \text{Min}: \quad & \|\boldsymbol{l}_t + \boldsymbol{r}_t\|_1 \\ \text{s.t.} \quad & -\boldsymbol{l}_t \leq 2(\boldsymbol{w}_t^\top \tilde{\boldsymbol{x}}_t)\tilde{\boldsymbol{x}}_t - \tilde{\boldsymbol{g}}_t \leq \boldsymbol{r}_t, \\ & \tilde{\boldsymbol{x}}_t = [\tilde{\boldsymbol{d}}_t, \epsilon, \delta]^\top, \tilde{\boldsymbol{d}}_t \in \{0,1\}^n, \ \boldsymbol{l}_t, \boldsymbol{r}_t \in \mathbb{R}^{n+2}. \end{aligned}$$

P_1 aims to minimize the ℓ_1 distance between $\boldsymbol{g}_t$ and $\tilde{\boldsymbol{g}}_t$. The optimization variables include $\tilde{\boldsymbol{d}}_t$, $\boldsymbol{l}_t$, and $\boldsymbol{r}_t$. The solution of $\tilde{\boldsymbol{d}}_t$ is the final noisy data. $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$ measure the bias and serve as auxiliary parameters for the collector to adjust biased gradient $\boldsymbol{g}_t$ caused by perturbed data $\tilde{\boldsymbol{d}}_t$ ($\tilde{\boldsymbol{x}}_t$) (Line 8 of Algorithm 2). $\tilde{\boldsymbol{x}}_t$, $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$ are then forwarded to the collector.

P_1 is a mixed integer quadratically constrained program (MIQCP) with the sum of absolute value as the objective function. Following [31], P_1 is transformed to the MIQCP with linear objective function, which can be optimally solved by plenty of commercial solvers, such as CPLEX.

Theoretical analysis of $\mathcal{A}_{r,I}$. As $\boldsymbol{g}_t$ is a biased gradient, the data collector cannot directly apply it to update $\boldsymbol{w}_t$. Instead, $\boldsymbol{g}'_t$, calculated as $\boldsymbol{g}'_t = \boldsymbol{g}_t - \boldsymbol{r}_t + \boldsymbol{l}_t$, is used in Algorithm 2. This is because $\boldsymbol{g}'_t$ becomes an unbiased gradient after adjustment, as stated in Lemma 3. We first present Lemma 2.

Lemma 2. *We have $\boldsymbol{g}'_t = \tilde{\boldsymbol{g}}_t = \boldsymbol{g}_t^* + \boldsymbol{Z}$, where $\boldsymbol{g}'_t = \boldsymbol{g}_t - \boldsymbol{r}_t + \boldsymbol{l}_t$.*

Since each element of $\boldsymbol{Z}$ follows the truncated Laplace distribution, the mean of $\boldsymbol{Z}$ is $\mathbf{0}$. Then we have $\mathbb{E}[\boldsymbol{g}_t^*|\boldsymbol{w}_t] = \mathbb{E}[\boldsymbol{g}'_t|\boldsymbol{w}_t]$. Besides, Proposition 2 claims that $\mathbb{E}[\boldsymbol{g}_t^*|\boldsymbol{w}_t] = \nabla F(\boldsymbol{w}_t)$. Hence, Lemma 3 directly follows.

Lemma 3. *The random variable $\boldsymbol{g}'_t$ satisfies $\mathbb{E}[\boldsymbol{g}'_t|\boldsymbol{w}_t] = \nabla F(\boldsymbol{w}_t)$.*

Aside from perturbed data, a data owner also forwards auxiliary parameters $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$ to the collector to facilitate the calculation of $\boldsymbol{g}'_t$. One concern is whether the disclosure of $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$ compromises owner's privacy. As proved in Lemma 4, $\mathcal{A}_{r,I}$ still satisfies (ϵ_I, δ)-LDP over the input, i.e., owner's raw data $\boldsymbol{x}_t$, even with the knowledge of $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$.

Lemma 4. *Randomized algorithm $\mathcal{A}_{r,I}$ satisfies (ϵ_I, δ)-LDP.*

The intuition why auxiliary parameters are nonsensitive. Auxiliary parameters, i.e., $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$, specify the bias, and thus are employed to do bias correction. The intuition why auxiliary parameters are nonsensitive is that

adversaries can only get noisy but unbiased gradients after the bias correction with those parameters. In other words, adversaries do not know genuine gradients, raw personal data, and decisions. Hence, adversaries can not identify any individual even with the knowledge of $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$.

Why is directly perturbing the raw data inapplicable? A key novelty of data perturbation in the exploration phase ($\mathcal{A}_{r,I}$) is that the noise is first inserted to the gradient which then generates perturbed data by solving an optimization problem. Conventional LDP mechanisms add noise to raw data directly, which are inapplicable here. As the gradient $\boldsymbol{g}_t$ derived from noisy data is biased with both approaches due to the nonlinearity of $\boldsymbol{g}_t$, the collector should remedy this bias properly. In MSGD, it is achieved by forwarding auxiliary parameters $\boldsymbol{r}_t$ and $\boldsymbol{l}_t$ to the collector. As proved in Lemma 4, $\mathcal{A}_{r,I}$ satisfies (ϵ_I, δ)-LDP even with the closure of auxiliary parameters. Under conventional LDP mechanisms, the collector should be aware of the bias denoted as $\boldsymbol{g}_t - \boldsymbol{g}_t^*$ so as to do bias correction. However, this value should not be revealed. Otherwise, $\boldsymbol{g}_t^*$ becomes public as it can be derived by subtracting $\boldsymbol{g}_t$ with the bias. As a result, user's private data $\boldsymbol{d}_t$ can be inferred as $\boldsymbol{g}_t^*$ is a function of $\boldsymbol{d}_t$. In a word, approaches of directly applying LDP for data perturbation and then employing bias correction is not viable in our scenario.

Why is directly submitting the perturbed gradients inapplicable? It is worth mentioning some prior works on SGD based differentially private learning [32], [33], [34], [35], [36], [37], [38], [39]. They enable the agent (the collector here) to learn owner's model $\boldsymbol{w}^*$ without viewing owner's data. They share the idea of adding noise to the gradients under the framework of SGD and forwarding noisy gradients to the agent, so that the model is gradually approximated at agent's side. These works are different from ours, as the agent only has access to the gradients but the data. Nonetheless, the collector is only interested in the data record itself rather "gradients" in real-world personal data trading markets [3], [4], [5], as data records have more usage than one specific query result (e.g., gradients). Now, one may suggest to provide to the collector the noisy data in addition to the noisy gradients. There is a limitation of this idea. Assume that the privacy budget is also ϵ_I. ϵ_I needs be divided into, say, ϵ_1 and ϵ_2 with $\epsilon_I = \epsilon_1 + \epsilon_2$. Then noise, generated following two probabilistic distributions under ϵ_1 and ϵ_2, is injected to the gradients and data, separately. The entire process satisfies ϵ_I-LDP according to *sequential composition* as stated in Theorem 1. Nonetheless, as a smaller privacy budget indicates more randomly distributed noise, the data produced via this approach is much more noisy and thus of less use to the collector, compared with $\mathcal{A}_{r,I}$.

4.2 Stage II: Decision Perturbation

Directly releasing the raw decision on whether to accept an offered price or not, i.e., $\mathbf{1}_{\{p_t \geq c_t\}}$, reveals c_t. Recall that c_t is a function of $\boldsymbol{d}_t$. Thus, to avoid the collector from inferring raw data $\boldsymbol{d}_t$ by analyzing data owner's trading decisions, we propose to perturb the decisions too. A randomized algorithm $\mathcal{A}_{r,II}$ is developed.

Definition 3. (Randomized algorithm $\mathcal{A}_{r,II}$). *Given the raw decision $\mathbf{1}_{\{p_t \geq c_t\}}$ as the input, the randomized algorithm $\mathcal{A}_{r,II}$ outputs the perturbed decision $a_t \in \{0, 1\}$.*

$\mathcal{A}_{r,II}$ follows the idea of *random response* (RR) [40]. It is a technique developed for the interviewees in a survey to give a random answer to a sensitive boolean question so that they can achieve plausible deniability. Specifically, each interviewee gives the raw answer with probability q and gives the opposite answer with probability $1 - q$. RR has been the predominant binary data perturbation mechanism for LDP. To satisfy ϵ-LDP, raw answer is revealed with probability $q = \frac{e^\epsilon}{e^\epsilon + 1}$ and the opposite answer is given with probability $1 - q = \frac{1}{e^\epsilon + 1}$. In our case, given the privacy parameter ϵ_{II} of $\mathcal{A}_{r,II}$, the probability that a data owner submits her raw decision is $q = \Pr[a_t = 1 | p_t \geq c_t] = \Pr[a_t = 0 | p_t < c_t] = \frac{e^{\epsilon_{II}}}{e^{\epsilon_{II}} + 1}$. The probability that she submits a false decision is $1 - q = \Pr[a_t = 0 | p_t \geq c_t] = \Pr[a_t = 1 | p_t < c_t] = \frac{1}{e^{\epsilon_{II}} + 1}$.

Lemma 5. *Randomized algorithm $\mathcal{A}_{r,II}$ satisfies ϵ_{II}-LDP.*

Discussions. Since an owner's trading decision is perturbed under $\mathcal{A}_{r,II}$, she is possible to accept an offer which produces negative payoff, i.e., $a_t = 1$ even $p_t < c_t$. Fortunately, as proved in Theorem 5, the owner's expected payoff, i.e., the difference between the payment and data cost, is nonnegative. Hence, they are well motivated to participate in data trading especially from a long-term view.

4.3 Piecing Together Stage I and Stage II

The perturbation $\mathcal{A}_r$ in the exploration phase consists of data perturbation $\mathcal{A}_{r,I}$ in stage I and decision perturbation $\mathcal{A}_{r,II}$ in stage II. As stated in Lemma 4 and Lemma 5, $\mathcal{A}_{r,I}$ and $\mathcal{A}_{r,II}$ satisfy (ϵ_I, δ)-LDP and ϵ_{II}-LDP (i.e., $(\epsilon_{II}, 0)$-LDP), respectively. Besides, $\mathcal{A}_{r,I}$ and $\mathcal{A}_{r,II}$ are deemed sequential operations as the execution of the latter relies on the output of the former. According to the *sequential composition* presented in Theorem 1, we have the following theorem intuitively.

Theorem 3. *$\mathcal{A}_r$ satisfies (ϵ, δ)-LDP, where $\epsilon = \epsilon_I + \epsilon_{II}$.*

5 Perturbation in Exploitation Phase

Like the exploration phase, perturbation in the exploitation phase $\mathcal{A}_t$ also consists of two stages: data perturbation in stage I $\mathcal{A}_{t,I}$ and decision perturbation in stage II $\mathcal{A}_{t,II}$. We aim to attain ϵ_I-LDP for $\mathcal{A}_{t,I}$ and ϵ_{II}-LDP for $\mathcal{A}_{t,II}$, where $\epsilon_I + \epsilon_{II} = \epsilon$.

Stage I: Data perturbation. In the exploitation phase, a data owner can trade Q pieces of data records in one round t. Let $\boldsymbol{d}_{t,q} \in \{0, 1\}^n$ be a single data record. The whole sets are represented as $\boldsymbol{d}_t = \{\boldsymbol{d}_{t,1}, \cdots, \boldsymbol{d}_{t,Q}\}$ and $\boldsymbol{x}_t = \{\boldsymbol{x}_{t,1}, \cdots, \boldsymbol{x}_{t,Q}\}$, where $\boldsymbol{x}_{t,q} = [\boldsymbol{d}_{t,q}, \epsilon, \delta]^\top$. Moreover, c_t is denoted as $c_t = \sum_{q=1}^{Q} \boldsymbol{w}^{*\top} \boldsymbol{x}_{t,q}$.

Definition 4. (Randomized algorithm $\mathcal{A}_{t,I}$). *Given the input $\boldsymbol{d}_t$, the randomized algorithm outputs the perturbed data $\tilde{\boldsymbol{d}}_t$.*

$\mathcal{A}_{t,I}$ follows the idea of RR. For $\boldsymbol{d}_{t,q}$, we perturb each bit sequentially under the privacy budget ϵ_I / n. The raw bit is revealed with probability $p = \Pr[\tilde{d}_{t,q} = 1 | d_{t,q} = 1] = \Pr[\tilde{d}_{t,q} = 0 | d_{t,q} = 0] = \frac{e^{\epsilon_I / n}}{e^{\epsilon_I / n} + 1}$. The opposite value is reported with probability $1 - p = \Pr[\tilde{d}_{t,q} = 1 | d_{t,q} = 0] = \Pr[\tilde{d}_{t,q} = 0 | d_{t,q} = 1] = \frac{1}{e^{\epsilon_I / n} + 1}$. Since $\boldsymbol{x}_{t,q}$ is assumed to be drawn

independently from a underlying distribution D, the bits are deemed disjoint. Therefore, given the parallel composition in Theorem 2, the randomized algorithm $\mathcal{A}_{t,I}$ satisfies ϵ_I-LDP.

Pricing rule. One remaining task in Algorithm 2 is to design a proper pricing rule p_t. As stated in Proposition 1, an ideal value is owner's data cost c_t. Recall that the collector derives $\boldsymbol{w}_{T_\alpha+1}$ in the last round of the exploration phase. Lemma 1 bounds its distance with $\boldsymbol{w}^*$ to a narrow margin. Still, p_t cannot be directly calculated by $\sum_{q=1}^{Q} \boldsymbol{w}_{T_\alpha+1}^\top \boldsymbol{x}_{t,q}$, as the raw data $\boldsymbol{d}_{t,q}$ ($\boldsymbol{x}_{t,q}$) is unavailable at the collector, but the perturbed data $\tilde{\boldsymbol{d}}_{t,q}$ ($\tilde{\boldsymbol{x}}_{t,q}$). Luckily, under the framework of RR, $\frac{p-1+\tilde{d}_{t,q}}{2p-1}$, a modification of $\tilde{d}_{t,q}$, is an unbiased estimate of $d_{t,q}$, as $\mathbb{E}[\frac{p-1+\tilde{d}_{t,q}}{2p-1}] = \mathbb{E}[d_{t,q}]$ [41]. Intuitively, we have $\mathbb{E}[\boldsymbol{x}'_{t,q}] = \mathbb{E}[\boldsymbol{x}_{t,q}]$, where $\boldsymbol{x}'_{t,q} = [\frac{(p-1)\boldsymbol{e}+\tilde{\boldsymbol{d}}_{t,q}}{2p-1}, \epsilon, \delta]^\top$ and $\boldsymbol{e} = \{1\}^n$. Then it is not difficult to infer $\mathbb{E}[\sum_{q=1}^{Q} \boldsymbol{w}_{T_\alpha+1}^\top \boldsymbol{x}'_{t,q}] = \mathbb{E}[\sum_{j=1}^{Q} \boldsymbol{w}_{T_\alpha+1}^\top \boldsymbol{x}_{t,q}]$. Finally, the pricing rule is published as

$$p_t = \sum_{q=1}^{Q} \boldsymbol{w}_{T_\alpha+1} \boldsymbol{x}'_{t,q} + \xi. \tag{1}$$

where

$$\xi = 2\sqrt{\frac{Q}{2}\log\frac{2}{\sigma'}} + Q\sqrt{(n+\epsilon^2+\delta^2)\frac{(624\log(\log(\tau)/\sigma)+1)G^2}{\lambda^2\tau}}.$$

ξ is used to compensate the difference between $\sum_{q=1}^{Q} \boldsymbol{w}_{T_\alpha+1} \boldsymbol{x}'_{t,q}$ and c_t caused by the gap between $\boldsymbol{w}_{T_\alpha+1}$ and $\boldsymbol{w}^*$. As shown in the proof of Theorem 6, the value of ξ plays a critical role in attaining sublinear regret of our scheme. The instantiation of σ and σ' is given in the proof of Theorem 6 too. The above expression is inserted back to line 15 of Algorithm 2. The pricing rule, i.e., parameters involved in (1) except $\tilde{\boldsymbol{x}}_t$, is first published by the collector. Then an owner calculates p_t based on her perturbed data $\tilde{\boldsymbol{x}}_t$ and decides if to accept the offer. If accepted, p_t is calculated by the collector based on the received data $\tilde{\boldsymbol{x}}_t$ and paid to the owner.

To ensure sublinear regret, this work makes an assumption on the value of Q.

Assumption 2. *The minimum number of data records Q that an owner sells to the collector at one time in the exploitation phase is at least $T^{1/3}$.*

In practice, if an owner does not meet the requirement at a specific time, she can choose to delay the trading, e.g., wait for a period of time until sufficient data records are collected.

Stage II: Decision Perturbation. $\mathcal{A}_{t,II}$ is the same with $\mathcal{A}_{r,II}$ for decision perturbation. Hence, ϵ_{II}-LDP directly follows. We thus omit its discussions here.

Piecing Together Stage I and Stage II. $\mathcal{A}_{t,I}$ and $\mathcal{A}_{t,II}$ satisfy ϵ_I-LDP and ϵ_{II}-LDP, respectively. Like the exploration phase, $\mathcal{A}_{t,I}$ and $\mathcal{A}_{t,II}$ are executed sequentially. Due to the *sequential composition* property, we have the following theorem intuitively.

Theorem 4. *$\mathcal{A}_t$ satisfies ϵ-LDP, where $\epsilon = \epsilon_I + \epsilon_{II}$.*

6 Theoretical Analysis

Learning performance of MSGD. In this part, we show that the proposed MSGD achieves sublinear regret.

Lemma 6. *Given any T such that $\tau \geq 4$, the proposed MSGD algorithm ensures $0 \leq p_t - c_t \leq 2\xi$, $\forall t \in [T_\alpha + 1, \bar{t}]$ with the probability at least $(1-\sigma)(1-\sigma')^2$, where the constants $\sigma, \sigma' \in (0, 1/e)$.*

It guarantees, with a high probability, $p_t \geq c_t$ in the exploitation phase. Based on Lemma 6, we further derive the following result.

Theorem 5. *For any T such that $\tau \geq 4$, with $\alpha = \sigma = \sigma' = T^{-1/3}$, owner's expected payoff, i.e., $\mathbb{E}[p_t - c_t]$, in each trading round $t \in [1, \bar{t}]$ is non-negative.*

This is a critical property for a self-sustained data trading market: An owner receives payment for selling her data no less than the incurred cost on average. Thus it is profitable for owners to join the market. On the other hand, it is undesirable for a collector to overpay for profit maximization. Lemma 6 bounds the overpay within 2ξ at a high probability. As discussed in the evaluation, the average regret per data record approximates to 0 as $T \to \infty$, so does the overpay per data record. As proved in Proposition 1, $R(T)$ is minimized for setting p_t equal to c_t. Due to the LDP-based privacy protection, it is extremely challenging to estimate c_t exactly based on owner's noisy inputs. Still, the following theorem, also the main result of this work, proves sublinear $R(T)$ attained by MSGD.

Theorem 6. *For any T such that $\tau \geq 4$, with $\alpha = \sigma = \sigma' = T^{-1/3}$, the MSGD algorithm has expected regret at most $R(T) = \mathcal{O}(T^{\frac{5}{6}}\sqrt{\log(T^{\frac{1}{3}})})$ which is sublinear.*

Discussion. In general, regret is with respect to the total number of iterations, i.e., $\bar{t}$ here. Instead, $R(T)$ is evaluated to the total number of data records T in this work. We make this compromise in trade of retaining sublinear $R(T)$. In our scenario, an owner is allowed to strategically report her decision without following the ground truth for privacy consideration. According to the *impossibility theory* proved in [29], [42], there is no online learning algorithm with sublinear regret, when entities' strategical behaviors are not weakened as training continues. To circumvent this, these works introduce the discount factor to the surplus function to retain sublinear regret. Instead of following their idea, we show that it is still possible to realize sublinearity, but for an alternative regret $R(T)$ that is respect to the total number of data records T.

Privacy protection. So far, we have proved that the perturbation in the exploration phase $\mathcal{A}_r$ and the exploitation phase $\mathcal{A}_t$ satisfies (ϵ, δ)-LDP (Theorem 3) and ϵ-LDP (Theorem 4), respectively. An owner can participate in multiple rounds of trading in both phases and sell her data newly generated in different periods. However, the owner is restricted from double selling. Thus, datasets traded in different rounds are disjoint to each other. By applying the *parallel composition* property in Theorem 2, the entire scheme achieves (ϵ, δ)-LDP as stated below.

Theorem 7. *Our scheme satisfies (ϵ, δ)-LDP.*

The privacy protection achieved by our scheme is on owner basis. For example, if two owners have different privacy budgets (ϵ, δ) and (ϵ', δ'), then our scheme provides (ϵ, δ)-LDP and (ϵ', δ')-LDP to each, separately.

Truthfulness. According to prior works [43], [44] that explore the relationship between differential privacy and mechanism design, differential privacy implicitly leads to approximate truthfulness, a relaxation of truthfulness where the incentive to misreport a value is non-zero, but tightly controlled (see Definition 5). The intuition behind this is that (local) differential private schemes guarantee their noisy outputs are insensitive to the change in the data of a single individual. Specifically, no matter how one data owner misreports her raw data (i.e., personal data and trading decisions in our paper), the noisy outputs of LDP scheme are almost the same. Note that the learned $\boldsymbol{w}_t$ and the price p_t depend on the noisy outputs, and thus are also almost the same. Therefore, there is no incentive for owners to misreport. The following is the formal definition of approximate truthfulness.

Definition 5. *(ϵ-dominant-strategy truthful [44]). A mechanism $\mathcal{M} : \mathcal{I}^n \to \mathcal{O}$ is ϵ-dominant-strategy truthful if, for all agents, all vectors of inputs I_{-t} and $I_t, I'_t \in \mathcal{I}$, $\mathcal{M}$ satisfies*

$$\mathbb{E}[u(I_t, \mathcal{M}(I_{-t}, I_t))] \geq \mathbb{E}[u(I_t, \mathcal{M}(I_{-t}, I'_t))] - \epsilon,$$

where u is the utility function: $\mathcal{I} \times \mathcal{O} \to [0, a]$ over the outcome space.

The following theorem proves our scheme achieves approximate truthfulness.

Theorem 8. *(Revised from Theorem 6 in [45]) Suppose the utility function u is bounded by $a > 0$. Let $\mathcal{M}$ be our scheme that is (ϵ, δ)-local differential private, where $\epsilon \leq 1$. Then, $\mathcal{M}$ is $2(\epsilon + \delta)a$-dominant-strategy truthful, where $a = 2$.*

7 Experimental Evaluation

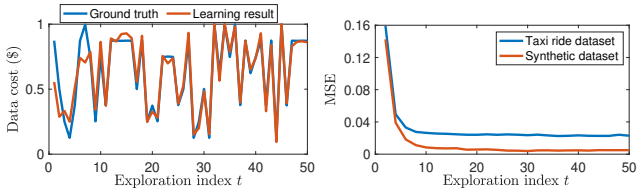
7.1 Data Preparation And Experiment Settings

Data preparation. We build our dataset from converting survey answers. The survey consists of questions regarding the volunteer's taxi rides recorded in all their ride-hailing apps, such as Uber and Lyft. Each trip record can be reviewed as a unique anonymous rider's response to a set of survey questions about his/her journey. Each question relates to a particular binary attribute of the ride. The attributes are coordinates/timestamps of pick-up/drop-off, payment method, trip distance, tip paid, toll paid, total fare, and passenger number as shown in Table 2. Besides, volunteers are asked to provide their privacy budgets $\epsilon, \delta \in [0, 1]$ toward each record, i.e., how sensitive they view attributes in the record, and data costs. For experiment simplicity, we adopt the same linear function that allows volunteers to derive their data cost based on their data and privacy budget. To allow one to estimate the implicit cost/value of some item, i.e, data cost here, there have been some mature solutions in the field of experimental economics [6], [23], [24]. As this part is out of the range of this work, we employ a simple method to have volunteers to estimate their data cost instead. In the three-month data collection campaign, we are able to collect more than $N = 18,000$ trip records from 30 volunteers. Note that the taxi ride dataset may not be strictly compliant with the linear cost model imposed by this work. As a control, we also generate a synthetic dataset by sampling a uniformly distributed set that satisfies linear cost model under a given $\boldsymbol{w}^*$.

TABLE 2
Questions regarding one taxi ride.

Attributes	Questions
CC usage	Do you pay with your credit card?
Toll	Does the trip involve toll?
Distance	Is the trip distance $\geq$ 10 miles?
Pick-up time	Is the pickup time $\geq$ 8PM?
Drop-off time	Is the drop-off time $\leq$ 3AM?
Pick-up location	Is the pick-up location within A city?
Drop-off location	Is the drop-off location within A city?
Tip	Do you pay the tip $\geq 25\%$ of the total fare?
Passenger #	Does passenger count > 1?
Fare	Is total fare $\geq$ \$10?

Experiment settings. To model the interaction between the collector and owners, we use PS-Lite [46], a light-weight implementation of the parameter-server framework to implement a simple distributed machine learning system containing two user nodes and one server node. Each user node contains Radeon Pro 570, Intel i5 processor, and 16GB memory. The server node contains Intel Iris Plus Graphics 655, Intel i7 processor, and 16GB memory. The communications among the nodes are established via a local Ethernet which has 1Gbps bandwidth. Moreover, the code for our mechanism, including MSGD and perturbation, is written in Python. CPLEX [47] is employed at the owner's side to solve optimization problems involved.

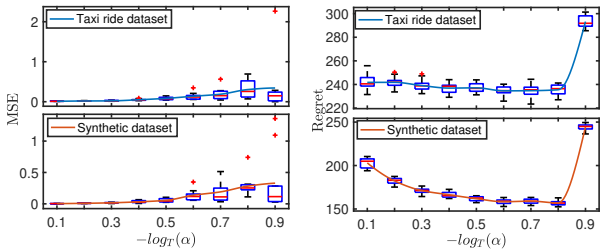


(a) Learning performance (with Taxi ride dataset) (b) MSE of learning results

Fig. 1. Learning performance over 50 trading rounds. ($\Delta_1/\epsilon_1 = 2$, $\delta = 10^{-4}$)

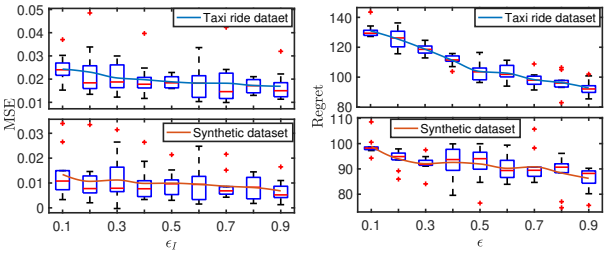
7.2 Experiment Results

Learning performance. We first evaluate the accuracy of our proposed MSGD in learning $\boldsymbol{w}^*$ in the exploration phase. Fig. 1(a) compares the estimate of owner's data cost c_t via MSGD and the ground truth over 50 transactions. We notice that the estimate follows the ground truth well after about 10 exploration rounds. This result complies with our theoretical result stated in Lemma 1; the gap between the learned parameter vector $\boldsymbol{w}_t$ and the target $\boldsymbol{w}^*$ is bounded and decreasing at a high probability as the trading continues. Fig. 1(b) further shows MSE of the learning result which measures the average of the squares of the learning errors. MSE drops quickly in the first a few rounds and then stays relatively stable and decreases slowly. Besides, we notice that MSE achieved under the Taxi ride dataset is slightly



(a) Impact of α on MSE (b) Impact of α on regret

Fig. 2. Impact of α on learning accuracy and learning regret. ($\Delta_1/\epsilon_1 = 5$, $\delta = 10^{-4}$, $T = 1000$)



(a) Impact of ϵ_I on MSE (b) Impact of ϵ on regret ($\epsilon_I = \epsilon_{II}$)

Fig. 3. Impact of privacy budget on learning accuracy and learning regret. ($\delta = 10^{-4}, T = 1000$)

above the one for the synthetic dataset. This is because the real-world dataset does not perfectly meet the linear model assumption in MSGD. Still, the gap of MSE under two datasets is pretty close and diminished as t increases.

Impact of α. We now examine the impact of α on the learning accuracy. Recall that α is a tunable parameter in MSGD, defined as the fraction of the exploration phase to the entire duration of T. The result of Fig. 2(a) is derived by varying $-\log_T(\alpha)$ from 0.1 to 0.9. For example, when $-\log_T(\alpha) = 0.1$, $\alpha T = T^{0.9}$ trading rounds are designated to exploration while the remaining tradings belong to the exploitation phase. According to the figure, MSE of both datasets drops to 0.015 almost linearly as $-\log_T(\alpha)$ decreases to 0.3. After that, the decrease of MSE slows down and approximate to 0.014 with a smaller $-\log_T(\alpha)$. This is because the convergence speed slows down as more training samples are fed in. Moreover, MSE of the taxi ride dataset is larger than that of the synthetic dataset due to the same reason mentioned above.

Impact of ϵ. We now examine the impact of privacy budget ϵ on the learning accuracy. Recall that ϵ_I controls the scale of inserted noise. The result of Fig. 3(a) is derived by varying ϵ_I from 0.1 to 0.9. MSE drops to 0.017 and 0.007 for taxi ride dataset and synthetic dataset, respectively, as ϵ_I increase to 0.9. This is because the larger ϵ_I produces less noise, which in turn brings up the learning accuracy and thus lower MSE.

Learning regret. One of the main goals in this work is to minimize the learning regret so as to maximize the collector's profit. The learning regret is expressed as $\sum_{t=1}^{\bar{t}}(r_t^* - r_t)$, i.e., the difference between the collector's maximum benefit and the actual one (by offering p_t) in each round t. We examine the regret with respect to $-\log_T(\alpha)$ through Fig. 2(b). The regret is accumulated over $T = 1000$ tradings records. We observe that the lowest regret of two datasets exists when $-\log_T(\alpha)$ is around 0.8. Either a too-small or a too-large $-\log_T(\alpha)$ leads to an enlarged regret. For the former, most offered prices are randomly generated and thus lead to poor profit; for the latter, insufficient rounds of tradings are devoted to exploration and thus result in poor learning performance. We observe a sharp decrease of regret as training rounds rise from $T^{0.1}$ to $T^{0.2}$ ($-\log_T(\alpha)$ decreases from 0.9 to 0.8). This is because the model in MSGD converges fast as shown in Fig.1(b) and becomes properly trained when training rounds reach $T^{0.2}$. Fig. 3(b) examines the regret with respect to ϵ. The value drops as the increase of ϵ for both datasets. This phenomenon can be explained from two aspects. In the exploration phase, a larger ϵ_I produces a smaller MSE as discussed above, which causes smaller regret. In the exploitation phase, an owner is more likely to reveal her true trading decision. Hence, the pricing rule derived via learning is more profitable. Fig. 4 depicts the average regret for each trading data record, expressed as $\sum_{t=1}^{\bar{t}}(r_t^* - r_t)/T$. It shows that average regret drops quickly when T increases to 500. Then it gradually approaches to 0 with a larger T.

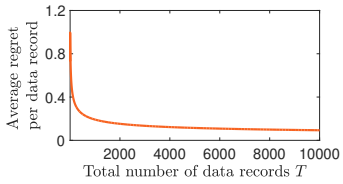
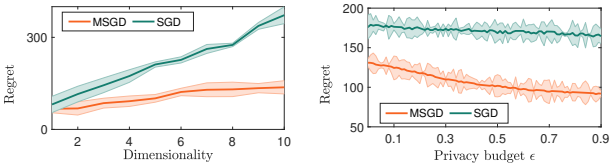


Fig. 4. Average regret for each trading data.

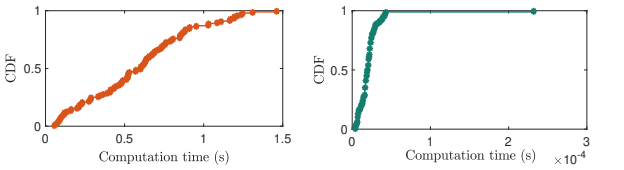


(a) Impact of attribute dimension n (b) Impact of privacy budget ϵ

Fig. 5. Regret comparison between MSGD and SGD.

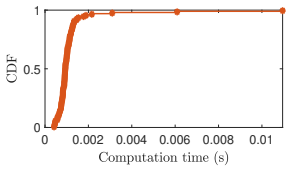
Comparison between MSGD and SGD. At the beginning of this work, we propose a naive learning approach based on SGD to derive optimum pricing. In Fig. 5, we compare the learning regret between this approach and our proposed MSGD. The solid lines represent the average value, while colored ribbons are min-max boundaries. Fig. 5(a) shows the regret with respect to attribute dimensions n. The regret grows linearly in general. Besides, MSGD outperforms SGD under all dimensions. This is because the naive algorithm cannot produce an unbiased estimate of gradient of F to update $\boldsymbol{w}_t$. As a result, the offered price can hardly lead to satisfactory regret. Fig. 5(b) shows the regret versus privacy budget ϵ. A small ϵ indicates a strict privacy requirement, i.e., the perturbed data is injected with a large amount of noise. Hence, a worse learning regret is exhibited. Still, MSGD outperforms SGD for producing $\boldsymbol{w}_t$ converging to $\boldsymbol{w}^*$ effectively.

Computation time. We examine the computation time for each trading of both the collector and data owners at exploration and exploitation phases in Fig. 6.

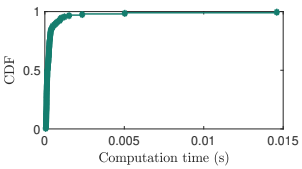


(a) Data owner, exploration phase

(b) Collector, exploration phase



(c) Data owner, exploitation phase



(d) Collector, exploitation phase

Fig. 6. Computation time for one trading.

TABLE 3
Communication overhead.

Dimensionality (n)	**2**	**4**	**6**	**8**	**10**
Date size (KB)	50	79	107	133	160
Time (ms)	31	49	65	81	98
Privacy budget ϵ	**0.9**	**0.7**	**0.5**	**0.3**	**0.1**
Data size (KB)	44	44	44	44	44
Time (ms)	27	28	27	27	28

As shown in Fig. 6(a), the computation time at the owner in the exploration phase ranges between 0.1 s and 1.5 s. Its average and 95th percentile are 0.5 s and 1.3 s, respectively. The most computation-demanding part is to solve P_1, a MIQCP. With the CPLEX solver, its optimum results can be derived efficiently. We further show in Fig. 6(c) the owner's computation time in the exploitation phase. Its average and 95th percentile are 1 ms and 3 ms, respectively. The significantly reduced computation time is due to lightweight noise generation and injection. Thus the perturbation mechanism developed for owners is practical to implement.

Fig. 6(b) plots the CDF of collector's computation time in the exploration phase. The dominant part is the update of $\boldsymbol{w}_t$. The value ranges from 10 μs to 250 μs, with the average and 95th percentile as 20 μs and 50 μs, respectively. Some extremely low values are observed due to the case of $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t$ when the owner rejects the offer. Fig. 6(d) plots the CDF of collector's computation time in the exploitation phase. Its average and 95th percentile is 1 ms and 5 ms, respectively. Since multiple data records are allowed to trade in each round of the exploitation phase, it experiences increased computation time compared with the exploration phase. Note that the collector is implemented on a desktop with mediocre hardware capacity in the experiments. An even lower time consumption will be observed by implementing the collector in a designated server or cloud platform. Besides, parallel computing can also be leveraged to enhance the computation performance.

Communication overhead. We then evaluate in Table 3 the transmitted payload data size between the collector and data owner for one transaction and the communication time caused thereby. The communication time takes into account the transmission delay and the propagation delay (negligible in local Ethernet). In the exploration phase, the transmitted data includes p_t (collector to owner), $\tilde{\boldsymbol{x}}_t$, $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$ (owner to collector). In the exploitation phase, the transmitted data include p_t (collector to owner), $\tilde{\boldsymbol{x}}_t$ containing Q records (owner to collector). Obviously, the transmitted data size in exploitation phase will be much larger, and thus causes a higher communication overhead. When $n = 6$ and $Q = 100$, the corresponding data size in the exploration phase is 107 KB transmitted within 65 ms. We observe a positive correlation between the communication overhead and the attribute dimension n. On the other hand, the privacy budget ϵ does not influence the communication overhead involved in each round. In general, the communication cost is relatively low in our mechanism.

8 Related Work

8.1 Pricing Data

Data pricing resides at the center of a data marketplace. It discusses how much to sell or how much to buy a piece of data or a dataset. The existing data pricing mechanisms can be generally categorized into query-based pricing and data-based pricing.

Query-based pricing. Papers in query-based pricing focus on query answers trading between the collector and data consumers. Works in this category basically answer the question of "how much to charge a data consumer for a query?". Essentially, the merchandise in the marketplace is query services based on the data rather than the data itself. There exist various mechanisms in this line of work (see [15], [48] for a survey), including a flat fee tariff, usage-based, and output-based. These pricing methods aim to provide market properties, e.g., arbitrage-freeness, profit maximization, etc. The vision for arbitrage-free query-based pricing is first introduced by Balazinska et al. [49], and is further developed in a series of papers [50], [51], [52], [53], [54], [55]. Here, arbitrage-freeness means that the data consumer can buy a query with a lower price than the market price by combining a bundle of other cheaper queries. Thus, the data collector needs to rule out arbitrage opportunities to preserve his revenue. Revenue maximization for query-based pricing is a relatively less explored area. Niu, Chawla, et al. [56], [57] study how the broker can maximize his revenue by posting reasonable prices for sequential queries. In parallel, the research community also proposes to use the notion of *privacy budget* to price data when data owners' privacy is taken into consideration [19], [22], [58], [59], [60], [61]. These mechanisms adopt the framework of *differential privacy*. To each query, differential privacy assigns the privacy budget ϵ that indicates how much information is leaked by the query. The data consumer then receives the noisy query result. Its charge is computed as a function of ϵ accordingly. At the same time, those mechanisms also achieve desired market properties, e.g., arbitrage-freeness, profit maximization, etc.

Our paper does not fall into the line of query-based pricing.

Data-based pricing. Papers in data-based pricing focus on (raw) data trading between the collector and data owners. We seek the answer to "how much to pay a data owner to compensate her privacy loss?". One category of prior

works [12], [13], [14], [15], [16], [22], [58], [59], [61] assumes a trustworthy collector who perturbs queries results and has access to all private information, e.g., raw data, data costs, etc. This line of work emphasizes to achieve various market properties. For example, fairness is the desired property where each data owner gets a fair share of the revenue in the coalition. Data owners are compensated by Shapley value to achieve fairness [14], [15], [16]. Compensations are carefully designed to achieve arbitrage-freeness [61]. Data owners are paid by data costs to guarantee the owner's non-negative payoff [12], [13], [58], [59]. A contract is carefully designed to achieve truthfulness of data owners reporting private information, etc [22]. The second category of prior works [17], [18], [19], [20], [21], [60], [62] assume an untrusted collector where data owners perturbed their data before selling it. In this line of work, as they aim to compensate data owners for the privacy loss, the payments are usually a function of privacy loss (also called data costs in this paper). Therefore, their problem is degraded to get data costs on the untrusted collector side. The paper [62] assumes data costs are public information. The other papers in the second category focus on the design of incentive mechanisms that truthfully elicit data costs on the untrusted collector side. Specifically, truthfulness mechanisms based on auctions [17], [18], [19], game theory [20], or contracts [21], [60] are proposed to elicit data owners to reveal her true cost honestly. All the works from the second category think data costs are not sensitive and thus are available from public information or incentive mechanisms. On the contrary, our paper thinks in addition to the owner's data, the data cost and their trading behaviors are also sensitive. Obviously, our paper considers a more challenging scenario.

8.2 Differentially Private Learning

Differentially private learning mainly studies constructing certain learning models from perturbed data. Based on where the data is perturbed (or whether the learner (collector here) is trusted), they are divided into two categories: local differentially private learning where noise is injected by the owner and differentially private learning where one trustworthy central learner, e.g., the repository where training data is stored, is responsible for data perturbation.

Empirical risk minimization in LDP has been studied in [38], [39], [63], [64], [65], [66]. In detail, [63], [64] focus on learning statistic characters, such as mean estimation and median estimation, of a given locally differentially private release of the dataset. They are targeting a different problem from ours. The remaining works aim to derive learning models by the empirical risk minimization with LDP as ours. Specifically, [65] approximates the loss function by perturbing intermediate results released by owners and solves the approximated loss function to get the noisy model. [38], [39], [66] are SGD based where the learner obtains noisy gradients from owners for constructing the model instead of receiving data, noisy or not. This line of work is inapplicable to data trading markets where the collector's primary goal is to purchase personal data instead of intermediate results. Details are given in section 4.1.

There are some prior works constructing private machine learning under the framework of standard differential privacy [32], [33], [34], [35], [36], [37], [67]. The noise is added in a centralized manner. Since we adopt local differential privacy framework, the noise is added at owners locally. Those techniques are inapplicable here.

9 Conclusions

Determining the price of personal data is of great importance for implementing the personal data market. In this paper, we study this problem in a setting where a data collector interacts with a set of data owners for their newly generated personal data. The goal of the collector is to pick proper prices that maximize his overall profit. Given that data owners perturb data and trading decisions for privacy protection, the task of data pricing becomes non-trivial. We cast the problem into an online stochastic optimization problem, by which the collector gradually constructs a model that captures the owner's data cost. To remedy the estimate error toward the optimal gradient caused by the noisy samples, we develop the MSGD algorithm that utilizes some auxiliary parameters to derive an unbiased estimation of the learning model. MSGD also attains sublinear regret of $\mathcal{O}(T^{\frac{5}{6}}\sqrt{\log(T^{\frac{1}{3}})})$. To facilitate the bias correction, we then modify the existing truncated Laplace based perturbation mechanism that satisfies (ϵ, δ)-LDP for owner's raw data and trading decisions. Experiment results show that our scheme achieves satisfactory learning accuracy with practical computation and communication overhead.

References

[1] F. Ferretti, *EU competition law, the consumer interest and data protection: The exchange of consumer information in the retail financial sector*, 2014.

[2] K. Laudon, "Markets and privacy," *Communs*, vol. 39, no. 9, p. 92, 1996.

[3] "Meeco," 2020, https://meeco.me/.

[4] "Datacoup," 2020, http://datacoup.com/.

[5] "Citizenme," 2020, https://www.citizenme.com/public/wp/about/.

[6] A. Acquisti, L. K. John, and G. Loewenstein, "What is privacy worth?" *The Journal of Legal Studies*, vol. 42, no. 2, pp. 249–274, June 2013.

[7] V. Shah, R. Johari, and J. Blanchet, "Semi-parametric dynamic contextual pricing," in *Proceedings of NIPS*, 2019.

[8] "Apple differential privacy technical overview," 2020, https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf.

[9] J. Gideon, L. Cranor, S. Egelman, and A. Acquisti, "Power strips, prophylactics, and privacy, oh my!" in *Proceedings of the ACM SOUPS*, 2006.

[10] E. Balistreri, G. McClelland, G. Poe, and W. Schulze, "Can hypothetical questions reveal true values? a laboratory comparison of dichotomous choice and open-ended contingent values with auction values," *Environmental and Resource Economics*, vol. 18, no. 3, pp. 275–292, 2001.

[11] X. Chen, D. Simchi-Levi, and Y. Wang, "Privacy-preserving dynamic personalized pricing with demand learning," *Management Science*, 2021.

[12] C. Niu, Z. Zheng, S. Tang, X. Gao, and F. Wu, "Making big money from small sensors: Trading time-series data under pufferfish privacy," in *Proceedings of the IEEE INFOCOM*, 2019.

[13] C. Niu, Z. Zheng, F. Wu, S. Tang, X. Gao, and G. Chen, "Unlocking the value of privacy: Trading aggregate statistics over private correlated data," in *Proceedings of the ACM SIGKDD*, 2018.

[14] J. Liu, J. Lou, J. Liu, L. Xiong, J. Pei, and J. Sun, "Dealer: an end-to-end model marketplace with differential privacy," *Proceedings of VLDB*, 2021.

[15] J. Pei, "A survey on data pricing: from economics to data science," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[16] R. C. Fernandez, P. Subramaniam, and M. J. Franklin, "Data market platforms: Trading data assets to solve data problems," *Proceedings of VLDB*, 2020.

[17] S. Zheng, Y. Cao, M. Yoshikawa, H. Li, and Q. Yan, "Fl-market: Trading private models in federated learning," *arXiv e-prints*, pp. arXiv–2106, 2021.

[18] L. Yang, M. Zhang, S. He, M. Li, and J. Zhang, "Crowd-empowered privacy-preserving data aggregation for mobile crowdsensing," in *Proceedings of the ACM Mobihoc*, 2018.

[19] A. Ghosh and A. Roth, "Selling privacy at auction," *Games and Economic Behavior*, vol. 91, pp. 334–346, May 2015.

[20] A. Fallah, A. Makhdoumi, A. Malekian, and A. Ozdaglar, "Optimal and differentially private data acquisition: Central and local mechanisms," *arXiv preprint arXiv:2201.03968*, 2022.

[21] Z. Zhang, S. He, J. Chen, and J. Zhang, "Reap: An efficient incentive mechanism for reconciling aggregation accuracy and individual privacy in crowdsensing," *Transactions on Information Forensics and Security*, vol. 13, no. 12, pp. 2995–3007, May 2018.

[22] L. K. Fleischer and Y.-H. Lyu, "Approximately optimal auctions for selling privacy when costs are correlated with data," in *Proceedings of the ACM EC*, 2012.

[23] "Wta," 2020, https://en.wikipedia.org/wiki/Willingness_to_accept.

[24] A. G. Winegar and C. R. Sunstein, "How much is data privacy worth? a preliminary investigation," *Journal of Consumer Policy*, vol. 42, no. 3, pp. 425–440, Sep 2019.

[25] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," 2014.

[26] G. Barthe, M. Gaboardi, J. Hsu, and B. Pierce, "Programming language techniques for differential privacy," *ACM SIGLOG News*, vol. 3, no. 1, pp. 34–53, Jan 2016.

[27] T. Jaksch, R. Ortner, and P. Auer, "Near-optimal regret bounds for reinforcement learning," *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1563–1600, April 2010.

[28] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," in *Proceedings of the ICML*, 2012.

[29] K. Amin, A. Rostamizadeh, and U. Syed, "Repeated contextual auctions with strategic buyers," in *Proceedings of NIPS*, 2014.

[30] Q. Geng, W. Ding, R. Guo, and S. Kumar, "Privacy and utility tradeoff in approximate differential privacy," *arXiv preprint arXiv:1810.00877*, 2018.

[31] "Transform an optimization with absolute value function to linear program," 2020, http://lpsolve.sourceforge.net/5.1/absolute.htm.

[32] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the ACM CCS*, 2016.

[33] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton, "Bolt-on differential privacy for scalable stochastic gradient descent-based analytics," in *Proceedings of the ACM SIGMOD*, 2017.

[34] P. Jain and A. Thakurta, "Differentially private learning with kernels," in *Proceedings of ICML*, 2013.

[35] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Proceedings of NIPS*, 2009.

[36] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *Proceedings of the IEEE GlobalSIP*, 2013.

[37] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, "cpsgd: Communication-efficient and differentially-private distributed sgd," in *Proceedings of NIPS*, 2018.

[38] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. C. Hui, H. Shin, J. Shin, and G. Yu, "Collecting and analyzing multidimensional data with local differential privacy," in *Proceedings of the IEEE ICDE*, 2019.

[39] T. T. Nguyên, X. Xiao, Y. Yang, S. C. Hui, H. Shin, and J. Shin, "Collecting and analyzing data from smart device users with local differential privacy," *arXiv preprint arXiv:1606.05053*, 2016.

[40] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the ACM CCS*, 2014.

[41] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *Symposium on USENIX Security*, 2017.

[42] K. Amin, A. Rostamizadeh, and U. Syed, "Learning prices for repeated auctions with strategic buyers," in *Proceedings of NIPS*, 2013.

[43] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proceedings of the IEEE International Symposium on Foundations of Computer Science*, 2007.

[44] L. Zhang, T. Zhu, P. Xiong, W. Zhou, and P. S. Yu, "More than privacy: Adopting differential privacy in game-theoretic mechanism design," *ACM Computing Surveys*, vol. 54, no. 7, pp. 1–37, Sep 2022.

[45] S. Leung and E. Lui, "Bayesian mechanism design with efficiency, privacy, and approximate truthfulness," in *International Workshop on Internet and Network Economics*, 2012.

[46] "Ps-lite," 2014, https://github.com/dmlc/ps-lite.

[47] "Cplex," 2020, https://www.ibm.com/products/ilog-cplex-optimization-studio.

[48] A. Muschalle, F. Stahl, A. Löser, and G. Vossen, "Pricing approaches for data markets," in *Workshop on BIRTE*, 2012.

[49] M. Balazinska, B. Howe, and D. Suciu, "Data markets in the cloud: An opportunity for the database community," *Proceedings of the VLDB*, 2011.

[50] P. Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu, "Query-based data pricing," *Journal of the ACM*, vol. 62, no. 5, pp. 1–44, Oct 2015.

[51] ——, "Querymarket demonstration: Pricing for online data markets," *Proceedings of the VLDB*, 2012.

[52] ——, "Toward practical query pricing with querymarket," in *Proceedings of the ACM SIGMOD*, 2013.

[53] S. Deep and P. Koutris, "Qirana: A framework for scalable query pricing," in *Proceedings of the ACM SIGMOD*, 2017.

[54] B.-R. Lin and D. Kifer, "On arbitrage-free pricing for general data queries," *Proceedings of the VLDB*, 2014.

[55] S. Deep, P. Koutris, and Y. Bidasaria, "Qirana demonstration: real time scalable query pricing," *Proceedings of the VLDB*, 2017.

[56] C. Niu, Z. Zheng, F. Wu, S. Tang, and G. Chen, "Online pricing with reserve price constraint for personal data markets," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[57] S. Chawla, S. Deep, P. Koutris, and Y. Teng, "Revenue maximization for query pricing," *arXiv preprint arXiv:1909.00845*, 2019.

[58] A. Ghosh and K. Ligett, "Privacy and coordination: computing on databases with endogenous participation," in *Proceedings of the ACM EC*, 2013.

[59] A. Ghosh, K. Ligett, A. Roth, and G. Schoenebeck, "Buying private data without verification," in *Proceedings of the ACM EC*, 2014.

[60] X.-B. Li and S. Raghunathan, "Pricing and disseminating customer data with privacy awareness," *Decision Support Systems*, vol. 59, pp. 63–73, March 2014.

[61] C. Li, D. Y. Li, G. Miklau, and D. Suciu, "A theory of pricing private data," *ACM Transactions on Database Systems*, vol. 39, no. 4, pp. 1–28, December 2014.

[62] W. Wang, L. Ying, and J. Zhang, "The value of privacy: Strategic data subjects, incentive mechanisms and fundamental limits," in *Proceedings of the ACM SIGMETRICS*, 2016.

[63] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Symposium on FOCS*, 2013.

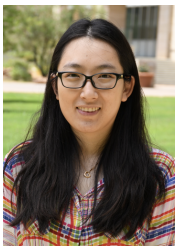
[64] ——, "Minimax optimal procedures for locally private estimation," *Journal of the American Statistical Association*, vol. 113, no. 521, pp. 182–201, May 2018.

[65] D. Wang, M. Gaboardi, and J. Xu, "Empirical risk minimization in non-interactive local differential privacy revisited," in *Proceedings of NIPS*, 2018.

[66] A. Smith, A. Thakurta, and J. Upadhyay, "Is interaction necessary for distributed private learning?" in *Symposium on S & P*, 2017.

[67] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: regression analysis under differential privacy," *Proceeding of VLDB*, 2012.

Mingyan Xiao received her M.S. degree from National University of Defense Technology in 2017, and B.E. degree from Nanjing University of Aeronautics and Astronautics in 2014, and the Ph.D degree in Computer Science from The University of Texas, at Arlington, in 2022, respectively. She is currently an assistant professor in the Department of Computer Science at California State Polytechnic University, Pomona. Her recent research interests are in the area of resource allocation and management in wireless networks, data-driven security and privacy.



Ming Li received the B.E. degree in Electrical Engineering from Sun Yat-sen University, China, in 2007, the M.E. degree in Electrical Engineering from Beijing University of Posts and Communications, China, in 2010, and the Ph.D. degree in Electrical and Computer Engineering from Mississippi State University, Starkville, in 2014, respectively. She is currently an associate professor in the Department of Computer Science and Engineering, The University of Texas at Arlington. Her research interests include mobile computing, internet of things, security, and privacy-preserving computing. Her work won Best Paper Awards in Globecom 2015 and DASC 2017, respectively. She received the NSF CAREER Award in 2020 and is a member of the IEEE.

Jennifer Jie Zhang is a professor of Information Systems and Fellow of the Eunice & James L. West Distinguished Professorship in the College of Business at the University of Texas at Arlington. She received her Ph.D. in computer information systems from the University of Rochester. She employs analytical and empirical techniques to closely examine issues in advanced and business applications of information technologies. Her research appears in MIS Quarterly, Journal of Economics and Management Strategies, Information Systems Research, Journal of Management Information Systems, Decision Support Systems, among others.