RPWithPrior: Label Differential Privacy in Regression

Haixia Liu *liuhaixia@hust.edu.cn*
School of Mathematics and Statistics & Institute of Interdisciplinary Research for Mathematics and Applied Science & Hubei Key Laboratory of Engineering Modeling and Scientific Computing, Huazhong University of Science and Technology, Wuhan, Hubei, China.

Ruifan Huang *ruifanhuang@hust.edu.cn*
School of Mathematics and Statistics & Institute of Interdisciplinary Research for Mathematics and Applied Science & Hubei Key Laboratory of Engineering Modeling and Scientific Computing, Huazhong University of Science and Technology, Wuhan, Hubei, China.

Reviewed on OpenReview: *https://openreview.net/forum?id=FiUeOOCMaj*

Abstract

With the wide application of machine learning techniques in practice, privacy preservation has gained increasing attention. Protecting user privacy with minimal accuracy loss is a fundamental task in the data analysis and mining community. In this paper, we focus on regression tasks under ϵ-label differential privacy guarantees. Some existing methods fundamentally convert a regression problem into a classification problem within the framework of Label Differential Privacy. However, such operations does not align well with real-world scenarios. To overcome these limitations, we model both original and randomized responses as *continuous* random variables, avoiding discretization entirely. Our novel approach estimates an optimal interval for randomized responses and introduces new algorithms designed for scenarios where a prior is either known or unknown. Additionally, we prove that our algorithm, RPWithPrior, guarantees ϵ-label differential privacy. Numerical results show that our method is always the best on the Communities and Crime. On Criteo Sponsored Search Conversion Log, and California Housing datasets, the performance of our approach remains comparable.

1 Introduction

Due to the widespread application of machine learning methods for model training, there has been a significant increase in attention towards the privacy of individual data in recent years. To address this concern and protect the privacy of data used for training, the concept of differential privacy was introduced as a popular notion by Dwork et al. (2006b).

Differential privacy (DP) aims to quantify the level of privacy protection provided by measuring the differences between two neighboring datasets and the privacy guarantees of a given mechanism. In the context of ϵ-DP, the parameter ϵ is commonly referred to as a privacy parameter or privacy budget. It serves as a knob to adjust the degree of privacy. Smaller values of ϵ correspond to stronger privacy guarantees, indicating a higher level of protection for individual data.

A particularly important class of differential privacy is known as label differential privacy (label DP). In label DP, the sensitive information lies in the labels, while the features are considered public and non-sensitive. For formal definitions of DP and Label DP, we refer the reader to the preliminary section in Appendix A.

In this work, we focus on Label DP. A canonical example of label DP is online advertising, where a user's profile and the contextual information of an ad are considered public features, while the ad conversion (the label or response) remains private. Additionally, demographic surveys can also be framed as label DP problems. In a census survey, the demographic information such as gender or age is considered public, while the income is deemed sensitive.

In the context of DP or label DP for classification problems, the labels are typically drawn from a finite set. However, in regression tasks, the response variable may not necessarily be limited to a finite number of values and could even be uncountable. Some existing methods (Ghazi et al., 2022; Badanidiyuru et al., 2023) fundamentally convert a regression problem into a classification problem within the framework of Label Differential Privacy. In this paper, we introduce a novel approach to tackle label differential privacy in regression from a distinct perspective. Our approach models both original and randomized responses directly as continuous random variables. We derive an optimal interval for randomized responses and propose a new algorithm designed to maximize the probability of preserving neighborhood relationships between original and randomized responses and guarantee ϵ-label differential privacy. Our contributions can be summarized as follows: (1) This work presents, to our knowledge, the first non-additive mechanisms for regression under the label DP model, operating directly on the continuous label space without conversion to classification. (2) We expand upon the RPWithPrior algorithm to address the scenario where the global prior distribution is unknown, which is also guarantees for ϵ-label DP. (3) We assess the performance of our method on three datasets.

The remainder of this paper is structured as follows: Section 2 reviews related works on differential privacy and label differential privacy in both classification and regression tasks. Section 3 presents our proposed method for achieving label differential privacy in regression with prior known, followed by the method for label differential privacy in regression with prior unknown in Section 4. Numerical implementations are discussed in Section 5. Finally, we conclude with our findings and highlight the limitations of this work in Section 6.

2 Related Works

Numerous methods exist for designing DP algorithms, including output or objective perturbation (Chaudhuri et al., 2011), DP versions of SGD (Song et al., 2013), functional mechanism (Zhang et al., 2012), DP-SGD with norm clipping and noise adding (Abadi et al., 2016), among others. One classical method predating the notion of differential privacy is randomized response (Warner, 1965), which was applied to inputs from a finite set. In the randomized responses (RR) algorithm, sensitive inputs are mapped to themselves with a certain probability and uniformly to other values with the remaining probability. Unfortunately, the RR algorithm suffers from a large error (Chan et al., 2012). Consequently, a significant number of works have been proposed to improve accuracy, often by relaxing to weaker privacy models (Acharya et al., 2020; Cheu et al., 2019; Erlingsson et al., 2019). The second class of differential privacy methods is the additive noise mechanisms. Popular mechanisms include the Gaussian mechanism (Dwork et al., 2006a; Smith et al., 2018; 2021), Laplace mechanism (Dwork et al., 2006b), staircase mechanism (Geng & Viswanath, 2014), and others, which add noise to adjacent inputs to calibrate sensitivity. Over the last decade, DP algorithms have been integrated with machine (deep) learning, applying to adversarial learning (Phan et al., 2020), multiple parties (Shokri & Shmatikov, 2015), language models (McMahan et al., 2017) and knowledge transfer (Papernot et al., 2016). For convenience, open-source libraries that integrate machine learning frameworks have been developed, such as TensorFlow Privacy and PyTorch Opacus.

For label Differential privacy, either classification or regression, numerous researchers have focused on this topic. In order to address the label differential privacy problem in classification, Gahzi et al. (2021) proposed the Label Privacy Multi-Stage Training (LP-MST) algorithm by calibrating the raw user data using the 'RRWithPrior' method, where the prior is predicted by the previous model.

Furthermore, Malek et al. (2021) presented two approaches, namely PATE and ALIBI, based on the PATE framework and the Laplace mechanism, specifically designed for label-privacy machine learning settings in classification. In the case of PATE, they utilized the PATE framework for semi-supervised learning, achieving an excellent tradeoff between empirical privacy loss and accuracy. On the other hand, ALIBI applied additive Laplace noise to a one-hot encoding of the label and then performed Bayesian inference using the prior to denoise the mechanism's output. They demonstrated that ALIBI improves upon the work of (Ghazi et al., 2021) in high-privacy regimes. In (Esfandiari et al., 2022), the authors presented new mechanisms for label differential privacy via clustering. Their approach involved clustering training examples based on non-private feature vectors, randomly resampling labels from examples within the same cluster, and producing a training set with noisy labels alongside a modified loss function. They demonstrated that if the clusters were large and of high quality, the model minimizing the modified loss converged to a small excess risk at a rate similar to non-private learning.

For regression tasks, one popular approach involves the additive noise mechanisms (Balle & Wang, 2018; Fernandes et al., 2021; Geng & Viswanath, 2014). A recent method is the RR-on-Bins mechanism (Ghazi et al., 2022), which discretized the output space into finite bins and applied the RR algorithm to these bins. The finite bins were determined using a dynamic programming approach. Another notable method is the optimal unbiased randomizers for regression with label differential privacy (Badanidiyuru et al., 2023), which introduced an unbiased constraint to enhance its performance.

3 Label DP for Regression with Prior Known

In this section, our focus is on regression with label differential privacy. We consider a dataset for regression denoted as $(x, y) \in (\mathcal{X}, \mathcal{Y})$, drawn from an unknown distribution. The regression task is to learn a predictor $f_\theta : \mathcal{X} \to \mathcal{Y}$ by minimizing a loss function $\mathcal{L}(f_\theta(x), y)$. The loss function, denoted as $\mathcal{L}(\cdot,\cdot)$, is typically used for regression tasks, such as mean squared error. Our goal is to design a randomized method that maps original responses to randomized responses by maximizing the probability of preserving neighborhood relationships, while ensuring ϵ-label differential privacy.

In this paper, we consider both the original and randomized response variables as continuous random variables. We denote the original response variable as Y and the corresponding randomized output variable as $\tilde{Y}$.

Let us consider the concept of ϵ-label differential privacy. We have the following proposition:

Proposition 3.1. *Let $\mathcal{A} : \mathcal{Y} \to \tilde{\mathcal{Y}}$ be a randomized algorithm and $f_{\tilde{Y}|Y}(\tilde{y}, y)$ represent the conditional probability density function conditioned on $Y = y$. $\mathcal{A}$ satisfying ϵ-label DP is equivalent to the following inequality:*

$$\int_{y_0}^{\tilde{y}} f_{\tilde{Y}|Y}(\tilde{y}, y)d\tilde{y} \le e^\epsilon \int_{y_0}^{\tilde{y}} f_{\tilde{Y}|Y}(\tilde{y}, y')d\tilde{y}, \ \forall y, y' \in \mathcal{Y}, \forall (y_0, \tilde{y}) \subseteq \tilde{\mathcal{Y}}. \tag{3.1}$$

When taking the derivative of both sides of (3.1) *with respect to $\tilde{y}$, we obtain $f_{\tilde{Y}|Y}(\tilde{y}, y) \le e^\epsilon f_{\tilde{Y}|Y}(\tilde{y}, y')$ holds for all $y, y' \in \mathcal{Y}$ and $\tilde{y} \in \tilde{\mathcal{Y}}$ almost everywhere.*

Remark 3.2. *In the following, we characterize ϵ-label DP as follows:*

$$f_{\tilde{Y}|Y}(\tilde{y}, y) \le e^\epsilon f_{\tilde{Y}|Y}(\tilde{y}, y'), \ \forall y, y' \in \mathcal{Y}, \forall \tilde{y} \in \tilde{\mathcal{Y}}. \tag{3.2}$$

Next, we formulate an optimization model to address regression problems under ϵ-label differential privacy with a known prior, as discussed in Subsection 3.1, using a fixed set $\mathcal{I}$. An optimal set $\mathcal{I}$ is determined in Subsection 3.2. Once the optimal set is established, a randomized algorithm is introduced, and the ϵ-label DP property of the proposed algorithm is ensured in Subsection 3.3.

3.1 Model Formulation

In this subsection, we assume we know the distribution of original responses of regression and the corresponding probability density function is $f_Y(y)$. Let $f_{\tilde{Y}|Y}(\tilde{y}, y)$ represent the conditional probability density function conditioned on $Y = y$ and $\mathcal{N}_y = [y - \zeta, y + \zeta]$ be a ζ-neighborhood of y, where ζ is a positive number.

Our method is motivated by the empirical distribution of the responses. Since most responses are concentrated within a finite interval, with only a few extreme outliers, we seek an optimal interval $\mathcal{I}$. This interval maximizes the probability that a response y is mapped to a randomized value $\tilde{y}$ within its neighborhood $\mathcal{N}_y$, for all $y \in \mathcal{I}$, while still satisfying ϵ-label differential privacy.

In this paper, we consider a subset $\mathcal{I} = [A_1, A_2] \subseteq (-\infty, +\infty)$ (determine in Subsection 3.2) and assume that $f_{\tilde{Y}|Y}(\tilde{y}, y)$ is a positive constant for any $y \in \mathcal{I}$ and $\tilde{y} \in \mathcal{N}_y$. Define $\mathcal{N}_\mathcal{I} = [A_1 - \zeta, A_2 + \zeta]$, our goal is to find a randomized algorithm mapping the original responses, whether in $\mathcal{I}$ or not, to the set $\mathcal{N}_{\mathcal{I}}$. This algorithm must ensure the ϵ-label DP property while maximizing the probability of transitioning from y to $\tilde{y} \in \mathcal{N}_y$ for all $y \in \mathcal{I}$. Then the optimization model is

$$\max \int_{\mathcal{I}} f_Y(y) \left[\int_{\mathcal{N}_y} f_{\tilde{Y}|Y}(\tilde{y}, y) d\tilde{y} \right] dy$$
$$\text{s.t.} \begin{cases} \int_{-\infty}^{+\infty} f_{\tilde{Y}|Y}(\tilde{y}, y) d\tilde{y} = 1, \forall y \in \mathcal{I}, \\ f_{\tilde{Y}|Y}(\tilde{y}, y) \leq e^{\epsilon} f_{\tilde{Y}|Y}(\tilde{y}, y'), \forall y, y' \in \mathcal{I}, \forall \tilde{y} \in \mathcal{N}_{\mathcal{I}}, \\ f_{\tilde{Y}|Y}(\tilde{y}, y) = v, \forall y \in \mathcal{I}, \forall \tilde{y} \in \mathcal{N}_y, \\ f_{\tilde{Y}|Y}(\tilde{y}, y) \geq 0, \forall \tilde{y}, y, \end{cases} \tag{3.3}$$

where the first and fourth constraints in (3.3) are from the properties of conditional probability density function, the second constraint in (3.3) is from (3.2) and the third constraint is from the assumption $f_{\tilde{Y}|Y}(\tilde{y}, y), \forall y \in \mathcal{I}, \forall \tilde{y} \in \mathcal{N}_y$ is a positive constant, denoted as v.

The following lemma shows the maximizer and the maximal value of the objective function in (3.3).

Lemma 3.3. *Let $\mathcal{I} = [A_1, A_2]$, ζ be a positive constant and $\epsilon > 0$ be a privacy budget. Define $\gamma = 2\zeta + e^{-\epsilon}(A_2 - A_1)$, $\mathcal{N}_y = [y - \zeta, y + \zeta], \forall y \in \mathcal{I}$ and $\mathcal{N}_{\mathcal{I}} = \cup_{y \in \mathcal{I}} \mathcal{N}_y$. Assume that $f_{\tilde{Y}|Y}(\tilde{y}, y)$ is a positive constant for any $y \in \mathcal{I}$ and $\tilde{y} \in \mathcal{N}_y$. For any $y \in \mathcal{I}$, the maximizer of* (3.3) *is*

$$f_{\tilde{Y}|Y}(\tilde{y}, y) = \begin{cases} 1/\gamma, & \text{if } \tilde{y} \in \mathcal{N}_y, \\ e^{-\epsilon}/\gamma, & \text{if } \tilde{y} \in \mathcal{N}_{\mathcal{I}}/\mathcal{N}_y, \\ 0, & \text{otherwise}, \end{cases} \tag{3.4}$$

and the maximal value of the objective function is

$$\int_{\mathcal{I}} f_Y(y) \left[\int_{\mathcal{N}_y} f_{\tilde{Y}|Y}(\tilde{y}, y) d\tilde{y} \right] dy \leq 2\frac{\zeta}{\gamma} \int_{\mathcal{I}} f_Y(y) dy = 2\frac{\zeta}{\gamma} \int_{A_1}^{A_2} f_Y(y) dy \triangleq F(A_1, A_2).$$

Proof. We defer the proof to Appendix B. □

It is important to note that the optimization model (3.3) can only provide $f_{\tilde{Y}|Y}(\tilde{y}, y)$ for all $y \in \mathcal{I}$. For $y \in (-\infty, +\infty) \setminus \mathcal{I}$, an ϵ-label DP property is also required. We summarize $f_{\tilde{Y}|Y}(\tilde{y}, y)$ for all y in Remark 3.4.

Remark 3.4. *In this paper, our goal is to find a randomized algorithm mapping original responses, whether in $\mathcal{I}$ or not, to the set $\mathcal{N}_{\mathcal{I}}$. To guarantee the ϵ-label DP property and combine* (3.4)*, the conditional probability density function $f_{\tilde{Y}|Y}(\tilde{y}, y), \forall y \notin \mathcal{I}$ requires to satisfy*

$$\max_{\tilde{y} \in \mathcal{N}_{\mathcal{I}}} (f_{\tilde{Y}|Y}(\tilde{y}, y)) \leq \frac{1}{\gamma}, \ \min_{\tilde{y} \in \mathcal{N}_{\mathcal{I}}} (f_{\tilde{Y}|Y}(\tilde{y}, y)) \geq \frac{e^{-\epsilon}}{\gamma}, \ f_{\tilde{Y}|Y}(\tilde{y}, y) = 0, \forall \tilde{y} \notin \mathcal{N}_{\mathcal{I}}, \ \int_{\tilde{y} \in \mathcal{I}} f_{\tilde{Y}|Y}(\tilde{y}|y) = 1. \tag{3.5}$$

One possible choice of $f_{\tilde{Y}|Y}(\tilde{y}, y), \forall y$ is

$$f_{\tilde{Y}|Y}(\tilde{y}, y) = \begin{cases} 1/\gamma, & \text{if } \tilde{y} \in \mathcal{N}_{P_{\mathcal{I}}(y)}, \\ e^{-\epsilon}/\gamma, & \text{if } \tilde{y} \in \mathcal{N}_{\mathcal{I}}/\mathcal{N}_{P_{\mathcal{I}}(y)}, \\ 0, & \text{otherwise}. \end{cases} \quad P_{\mathcal{I}}(y) = \begin{cases} A_2, & y > A_2, \\ y, & y \in \mathcal{I}, \\ A_1, & y < A_1. \end{cases} \tag{3.6}$$

Note that $\epsilon \geq 0$, then $e^{-\epsilon} \leq 1$ and $\gamma = 2\zeta + e^{-\epsilon}(A_2 - A_1) \leq 2\zeta + (A_2 - A_1)$, $e^{\epsilon}\gamma = 2\zeta e^{\epsilon} + (A_2 - A_1) \geq 2\zeta + (A_2 - A_1)$. Therefore, $e^{-\epsilon}/\gamma = 1/(e^{\epsilon}\gamma) \leq 1/(2\zeta + (A_2 - A_1)) \leq 1/\gamma$. Hence

$$f_{\tilde{Y}|Y}(\tilde{y}, y) = \begin{cases} 1/\gamma, & \text{if } y \in \mathcal{I}, \tilde{y} \in \mathcal{N}_y, \\ e^{-\epsilon}/\gamma, & \text{if } y \in \mathcal{I}, \tilde{y} \in \mathcal{N}_{\mathcal{I}}/\mathcal{N}_y, \\ \frac{1}{2\zeta + (A_2 - A_1)}, & \text{if } y \notin \mathcal{I}, \tilde{y} \in \mathcal{N}_{\mathcal{I}}, \\ 0, & \text{otherwise}. \end{cases} \tag{3.7}$$

Equations (3.6) and (3.7) present two options for $f_{\tilde{Y}|Y}(\tilde{y}, y)$ as discussed in Remark 3.4. Other choices are acceptable as long as the condition (3.5) is satisfied.

3.2 The Optimal Interval

Algorithm 1 Compute Optimal Interval

Require: A response $n_0 < n_1 < \cdots < n_k$, $\alpha_0, \cdots, \alpha_{k-1}$, $f = 0$.

for $i = 0$ **to** $k-1$ **do**
 for $j = i$ **to** $k-1$ **do**
  $l_1 = n_i$, $l_2 = n_{i+1}$, $m_1 = n_j$ and $m_2 = n_{j+1}$,
  if $f < \max_{p,q=1,2}(F(l_p, m_q))$ **then** ▷ Check $A_1 = n_i$ or n_{i+1}, $A_2 = n_j$ or n_{j+1}.
   $f = \max_{p,q=1,2}(F(l_p, m_q))$, $[A_1, A_2] = \arg\max_{p,q=1,2}(F(l_p, m_q))$.
  end if
  Compute possible critical points $A_1 = e_1$, $A_2 = e_2$.
  if $n_j < e_2 < n_{j+1}$ **then** ▷ if $e_2 \in [n_j, n_{j+1}]$
   if $f < \max(F(n_i, e_2), F(n_{i+1}, e_2))$ **then** ▷ check $A_1 = n_i$ or n_{i+1}, $A_2 = e_2$
    $f = \max(F(n_i, e_2), F(n_{i+1}, e_2))$,
    $[A_1, A_2] = \arg\max(F(n_i, e_2), F(n_{i+1}, e_2))$.
   end if
  end if
  if $n_i < e_1 < n_{i+1}$ **then** ▷ if $e_1 \in [n_i, n_{i+1}]$
   if $f < \max(F(e_1, n_j), F(e_1, n_{j+1}))$ **then** ▷ check $A_1 = e_1, A_2 = n_j$ or n_{j+1}
    $f = \max(F(e_1, n_j), F(e_1, n_{j+1}))$,
    $[A_1, A_2] = \arg\max(F(e_1, n_j), F(e_1, n_{j+1}))$.
   end if
  end if
  if $n_j < e_2 < n_{j+1}$ **and** $n_i < e_1 < n_{i+1}$ **then** ▷ if $e_2 \in [n_j, n_{j+1}]$ and $e_1 \in [n_i, n_{i+1}]$,
   if $f < F(e_1, e_2)$ **then** ▷ check $A_1 = e_1$, $A_2 = e_2$
    $A_1 = e_1$, $A_2 = e_2$.
   end if
  end if
 end for
end for
return A_1 and A_2.

For a given interval $[A_1, A_2]$, Lemma 3.3 gives the maximum achievable value $F(A_1, A_2)$ of the optimization problem (3.3). Our goal is to find the optimal interval by maximizing $F(A_1, A_2)$ over all possible A_1 and A_2. Next, we estimate an optimal choice for the interval $\mathcal{I} = [A_1, A_2]$ based on an available prior.

In practice, a simple technique for estimating a prior is a histogram, which corresponds to a piecewise constant function. Here we assume that $f_Y(y)$ is a step function with nodes $n_0 < n_1 < \cdots < n_k$ and analyze the scenarios in which A_1 and A_2 fall into these $k+2$ intervals with $A_1 \le A_2$, respectively.

The following lemma aims to identify the maximizers (A_1, A_2) of the objective function $F(A_1, A_2)$.

Lemma 3.5. *Suppose the support of $f_Y(y)$ consists of k intervals with endpoints $n_0 < n_1 < \cdots < n_k$ and*

$$f_Y(y) = \begin{cases} \alpha_i, & y \in [n_i, n_{i+1}), i = 0, \cdots, k-2, \\ \alpha_{k-1}, & y \in [n_{k-1}, n_k], \\ 0, & \textit{otherwise.} \end{cases}$$

Let ζ be a positive constant, $\gamma = 2\zeta + e^{-\epsilon}(A_2 - A_1)$ and $F(A_1, A_2) = \frac{2\zeta}{\gamma}\int_{A_1}^{A_2} f_Y(y)dy$, where $F(A_1, A_2)$ is the maximum achievable value of the optimization problem (3.3) *(derived in Lemma 3.3). When $A_1 \in (-\infty, n_0)$, $F(A_1, A_2)$ is increasing about A_1. When $A_2 \in (n_k, \infty)$, $F(A_1, A_2)$ is decreasing about A_2. When*

$A_1 \in [n_i, n_{i+1}]$ and $A_2 \in [n_j, n_{j+1}]$ with $i \le j$,

$$\nabla F_{A_1}(A_1, A_2) = \frac{-2\zeta}{\gamma^2}(e^{-\epsilon}(\alpha_i - \alpha_j)A_2 + c_1), \nabla F_{A_2}(A_1, A_2) = \frac{2\zeta}{\gamma^2}(e^{-\epsilon}(\alpha_i - \alpha_j)A_1 + c_2),$$

and the possible critical points are $A_1 = e_1 = \frac{c_2}{e^{-\epsilon}(\alpha_j - \alpha_i)}$ and $A_2 = e_2 = \frac{c_1}{e^{-\epsilon}(\alpha_j - \alpha_i)}$, where $h = \alpha_i n_{i+1} + \sum_{\ell=i+1}^{j-1} \alpha_\ell (n_{\ell+1} - n_\ell) - \alpha_j n_j$, $c_1 = 2\zeta\alpha_i - e^{-\epsilon}h$, $c_2 = 2\zeta\alpha_j - e^{-\epsilon}h$.

Proof. The proof defers to Appendix C. ☐

From Lemma 3.5, we find $F(A_1, A_2)$ increases when $A_1 \in (-\infty, n_0)$ and $F(A_1, A_2)$ decreases when $A_2 \in (n_k, +\infty)$, so the maximizer must be in the interval $[n_0, n_k]$. We turn to enumerate the k intervals in $[n_0, n_k]$. When $A_1 \in [n_i, n_{i+1}]$ and $A_2 \in [n_j, n_{j+1}]$ with $i \le j$, the possible critical points are $A_1 = e_1$ and $A_2 = e_2$.

To determine the maximizer, it is sufficient to evaluate the endpoints and points where the partial derivatives are zero. It is important to note that we will only check the value of e_1 if it falls within the interval $[n_i, n_{i+1}]$. Similarly, we will only check the value of e_2 if it falls within the interval $[n_j, n_{j+1}]$. The algorithm to compute optimal interval is described in Algorithm 1.

In Algorithm 1, the memory complexity is $O(k)$ because we only need to store a constant number of variables, including $n_0, \cdots, n_k$, $\alpha_0, \cdots, \alpha_{k-1}$, A_1, A_2, f, e_1 and e_2, where k represents the number of intervals with piecewise constants in the support. The time complexity of Algorithm 1 is at most $O(k^2)$. In each iteration, we need to evaluate the function F, which requires $O(1)$ operations. We iterate over $i = 0, \cdots, k-1$ and $j = i, \cdots, k-1$. Generally, k is not large, which can be controlled by users. Therefore, the algorithm for computing the optimal interval is efficient.

3.3 Response Privacy with Prior

Once the optimal interval $\mathcal{I}$ is chosen, the randomized output can be assigned to a specific point within $\mathcal{N}_\mathcal{I}$ according to the conditional probability density function (3.4), if the input response falls within this interval $\mathcal{I}$. Otherwise, we sample according to the conditional probability density function $f_{\tilde{Y},Y}(\tilde{y}, y)$ satisfying (3.5). The whole procedure is presented in Algorithm 2, named as *RPWithPrior*, to sample the randomized response $\tilde{y}$.

Algorithm 2 Response Privacy with Prior (RPWithPrior$_\epsilon$)

Require: A response y, a positive value ζ, privacy budget ϵ.
1: Compute the optimal A_1, A_2 by Algorithm 1.
2: **if** $y \in [A_1, A_2]$ **then**
3:     Sample $\tilde{y}$ from the distribution with (3.4),
4: **else**
5:     Sample $\tilde{y}$ according to $f_{\tilde{Y}|Y}(\tilde{y}, y)$ satisfying (3.5).
6: **end if**
7: **return** A randomized response $\tilde{y}$.

Finally, we will show *RPWithPrior$_\epsilon$ is ϵ-label DP.*

Theorem 3.6. *RPWithPrior is ϵ-label DP.*

Proof. Note that $f_{\tilde{Y}|Y}(\tilde{y}, y)$ is represented by (3.4) when $y \in \mathcal{I}$, otherwise $f_{\tilde{Y}|Y}(\tilde{y}, y)$ satisfies (3.5). For all y, we have $f_{\tilde{Y}|Y}(\tilde{y}, y) = 0$, when $\tilde{y} \notin \mathcal{N}_\mathcal{I}$. This means the randomized response $\tilde{y} \in \mathcal{N}_\mathcal{I}$ whenever the original response is.

For any $\tilde{y} \in \mathcal{N}_\mathcal{I}$ and any y, $\max(f_{\tilde{Y}|Y}(\tilde{y}, y)) = \frac{1}{\gamma}$ and $\min(f_{\tilde{Y}|Y}(\tilde{y}, y)) = \frac{e^{-\epsilon}}{\gamma}$, which satisfies the ϵ-label DP property. Therefore, we can conclude the result. ☐

4 Label DP for Regression with Prior Unknown

Our approach, RPWithPrior, is based on the availability of priors that are known in advance. However, there are scenarios where obtaining such priors is not always feasible. In this section, we propose an alternative algorithm that uses a histogram to approximate the prior $f_Y(y)$, computed from randomized responses generated by an additive mechanism (e.g., the Laplace mechanism) to guarantee label DP.

Let's consider the training dataset denoted as D and $\mathcal{M}_{\epsilon,\mathrm{Lap}}(y)$ be the randomized response of y by Laplace mechanism for any $(x,y) \in D$. To be precise, we first calculate the sample expectation μ: $\mu = \frac{1}{\#(D)} \sum_{(x,y)\in D} \mathcal{M}_{\epsilon,\mathrm{Lap}}(y)$, where $\#(\cdot)$ is the element number in the set. For a given positive parameter σ, let k_0 and k_1 be integers satisfying $\mu + k_0\sigma \le \min_{(x,y)\in D} \mathcal{M}_{\epsilon,\mathrm{Lap}}(y) < \mu + (k_0+1)\sigma$ and $\mu + k_1\sigma < \max_{(x,y)\in D} \mathcal{M}_{\epsilon,\mathrm{Lap}}(y) \le \mu + (k_1+1)\sigma$, the nodes are set as $n_0 = \min_{(x,y)\in D} \mathcal{M}_{\epsilon,\mathrm{Lap}}(y)$, $n_i = \mu + (i+k_0)\sigma, i = 1, \cdots, k_1 - k_0$ and $n_{k_1-k_0+1} = \max_{(x,y)\in D} \mathcal{M}_{\epsilon,\mathrm{Lap}}(y)$. Define $[n] = \{0, \cdots, n\}$, $\forall n \in \mathbb{N}$ and $J_k = [n_k, n_{k+1})$, if $k \in [k_1 - k_0 - 1]$, and $J_k = [n_k, n_{k+1}]$, if $k = k_1 - k_0$, then the histogram expression of $f_Y(y)$ is

$$f_Y(y) = \begin{cases} \frac{\#(S_k)}{\#(D)}, & y \in J_k, \forall k \in [k_1 - k_0], \\ 0, & \text{otherwise}, \end{cases} \quad S_k = \{y | \mathcal{M}_{\epsilon,\mathrm{Lap}}(y) \in J_k, (x,y) \in D\}.$$

The histogram-based algorithm for prior estimation is then formulated as follows:

Algorithm 3 Estimate the prior by histogram (Hist_ϵ)

Require: Dataset D, a positive parameter $\sigma > 0$.

1: $\mu = \frac{1}{\#(D)} \sum_{(x,y)\in D} \mathcal{M}_{\epsilon,\mathrm{Lap}}(y)$, ▷ Sample mean

2: Compute integers k_0 and k_1 satisfying

$$\mu + k_0\sigma \le \min_{(x,y)\in D} \mathcal{M}_{\epsilon,\mathrm{Lap}}(y) < \mu + (k_0+1)\sigma, \ \mu + k_1\sigma < \max_{(x,y)\in D} \mathcal{M}_{\epsilon,\mathrm{Lap}}(y) \le \mu + (k_1+1)\sigma.$$

3: Compute nodes of histogram

$$n_i = \begin{cases} \min_{(x,y)\in D} \mathcal{M}_{\epsilon,\mathrm{Lap}}(y), & \text{if } i = 0, \\ \mu + (i+k_0)\sigma, & \text{if } i = 1, \cdots, k_1 - k_0, \\ \max_{(x,y)\in D} \mathcal{M}_{\epsilon,\mathrm{Lap}}(y), & \text{if } i = k_1 - k_0 + 1. \end{cases}$$

4: Compute intervals of histogram

$$J_k = \begin{cases} [n_k, n_{k+1}), & k \in [k_1 - k_0 - 1], \\ [n_k, n_{k+1}], & k = k_1 - k_0, \end{cases} \quad S_k = \{y | \mathcal{M}_{\epsilon,\mathrm{Lap}}(y) \in J_k, (x,y) \in D\}.$$

5: $\alpha_k = \#(S_k)/\#(D), k = 0, \cdots, k_1 - k_0$. ▷ Histogram expression

6: **return** $\{(J_k, \alpha_k)\}$.

Remark 4.1. *The length of the histogram intervals in Algorithm 3 can be adjusted according to specific requirements by modifying the value of σ. When the prior distribution is complex, a smaller interval length enables the histogram to more accurately approximate the prior distribution; however, this comes at the cost of increased computational complexity. In practice, it suffices to choose a simple prior with a larger σ, because the data used to estimate the prior is perturbed by additive noise. A complex prior would risk learning this unwanted noise information.*

By utilizing the histogram as the prior, we can employ Algorithm 2 to obtain the optimal values of A_1 and A_2. These optimal values are then used to sample for each (x,y) in the dataset D. Subsequently, we train the model M on the datasets. To provide a comprehensive overview, we present the complete algorithm in Algorithm 4.

Algorithm 4 Response Privacy with Histogram

Require: Dataset $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ and privacy budgets ϵ_1, ϵ_2.

Set $\tilde{D} = \emptyset$,
estimate the prior by histogram via $\mathrm{Hist}_{\epsilon_1}$ in Algorithm 3,
for $(x_i, y_i) \in D$ **do**
  sample $\tilde{y}_i$ according to $\mathrm{RPWithPrior}_{\epsilon_2}$ Algorithm 2,
  $\tilde{D} = \tilde{D} \cup \{(x_i, \tilde{y}_i)\}$,
end for
M is the regression model trained on the set $\tilde{D}$.
return M.

Theorem 4.2. *Algorithm 4 is ϵ-label DP.*

Proof. Algorithm 4 splits the privacy budget into ϵ_1, ϵ_2, which follows that the entire algorithm is $(\epsilon_1 + \epsilon_2)$-label DP. ☐

5 Numerical Implementation

In numerical experiment section, we conduct a comprehensive performance evaluation of our proposed method and compare it with several existing mechanisms, including the Gaussian mechanism (Dwork et al., 2006a), the Laplace mechanism (Dwork et al., 2006b), the staircase mechanism (Geng & Viswanath, 2014), RRonBins (Ghazi et al., 2022) and Unbiased (Badanidiyuru et al., 2023). To avoid overfitting, we incorporate an L_2 regularizer into the loss functions of all these mechanisms. It's worth noting that the Gaussian mechanism is an approximate differential privacy method, and as such, we cannot set $\zeta = 0$ (as defined in the DP framework) in our numerical tests. Here, we set $\zeta = 10^{-4}$ in the following experiments.

Our focus is on three specific datasets: the Communities and Crime dataset (Redmond & Baveja, 2002), the Criteo Sponsored Search Conversion Log dataset (Tallis & Yadav, 2018), and the California housing dataset (Pace & Barry, 1997). In the following experiments, we conduct a total of 10 trials for the Gaussian, Laplace, Staircase, RRonBins mechanisms and our proposed mechanism. In each trial, we randomly divide the entire dataset into training (80%) and test (20%) sets using different random seeds. In addition, we use the overall privacy budget for fair comparison of algorithms that operate with or without a prior.

For all regression tasks in our experiments, we employ mean squared error (MSE) as the loss function $\mathcal{L}$ and evaluate performance using both the mean and standard deviation of the test MSE. The reported results are the average and standard deviation of the test mean squared error over the 10 trails. Note that we evaluate results based on the interval ranging from mean minus standard deviation to mean plus standard deviation. The best-performing intervals are those that overlap with the interval of the smallest mean, which are highlighted in bold. All experiments were conducted on a workstation equipped with an Intel Xeon W-2245 CPU, NVIDIA RTX 3080 Ti GPU, and 128GB RAM, using PyTorch under CUDA 11.7. The numerical implementations are available at `https://github.com/liuhaixias1/Response_privacy/`. In addition, parameters, computational complexity analyses are provided in Appendix D.

5.1 The Communities and Crime Dataset

The Communities and Crime dataset is a combined socio-economic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR, which can be downloaded from the website of UCI machine learning repository[1]. The creator is Michael Redmond from La Salle University. It is multivariate with 1994 instances and 128 attributes. Note that there are some missing values in the dataset, it is necessary to deal with in advance. In this paper, we use the similar technique as the paper of Selective Regression Under Fairness Criteria (Shah et al., 2022). First of all, we remove the useless attributes for regression, including *state, county, community, communityname, fold.*

[1] `https://archive.ics.uci.edu/ml/datasets/communities+and+crime`

Table 1: Comparison results on the Communities and Crime dataset.

Privacy Budget	Laplace	Gaussian	Staircase	RRonBins	Ours	Unbiased
	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std
0.05	8.3956 ± 3.1062	33.0362 ± 11.6948	7.0269 ± 3.0227	0.1043 ± 0.0227	**0.0643 ± 0.0076**	3.9640 ± 1.6592
0.1	2.4534 ± 1.1036	17.5311 ± 7.1242	2.3541 ± 0.6848	0.0918 ± 0.0139	**0.0621 ± 0.0054**	1.4708 ± 0.4732
0.3	0.3232 ± 0.0631	2.4218 ± 1.1848	0.2973 ± 0.0731	0.0940 ± 0.0209	**0.0552 ± 0.0022**	0.1808 ± 0.0649
0.5	0.1615 ± 0.0513	0.8934 ± 0.3259	0.1127 ± 0.0226	0.0892 ± 0.0254	**0.0507 ± 0.0051**	0.0732 ± 0.0174
0.8	0.0679 ± 0.0122	0.4376 ± 0.1317	0.0621 ±0.0124	0.0860 ± 0.0111	**0.0416 ± 0.0033**	**0.0376 ± 0.0077**
1.0	0.0533 ± 0.0117	0.3138 ± 0.1096	0.0467 ± 0.0064	0.0794 ± 0.0140	**0.0370 ± 0.0036**	**0.0334 ± 0.0036**
1.5	**0.0353 ± 0.0081**	0.1481 ± 0.0481	**0.0344 ± 0.0055**	**0.0281 ± 0.0039**	**0.0304 ± 0.0031**	**0.0279 ± 0.0034**
2	**0.0275 ± 0.0032**	0.0947 ± 0.0167	**0.0255 ± 0.0027**	0.0467 ± 0.0087	**0.0300 ± 0.0035**	**0.0241 ± 0.0031**
3	**0.0223 ± 0.0018**	0.0600 ± 0.0132	**0.0241 ± 0.0026**	0.0302 ± 0.0050	**0.0220 ± 0.0018**	**0.0228 ± 0.0024**
4	**0.0216 ± 0.0020**	0.0368 ± 0.0084	**0.0239 ± 0.0033**	**0.0245 ± 0.0020**	**0.0197 ± 0.0020**	**0.0226 ± 0.0017**
6	**0.0188 ± 0.0017**	0.0291 ± 0.0042	0.0249 ± 0.0025	**0.0191 ± 0.0025**	**0.0198 ± 0.0020**	**0.0213 ± 0.0024**
8	**0.0201 ± 0.0021**	**0.0231 ± 0.0020**	**0.0241 ± 0.0024**	**0.0195 ± 0.0024**	**0.0192 ± 0.0017**	**0.0203 ± 0.0009**
$+\infty$	**0.0182 ± 0.0016**	**0.0182 ± 0.0016**	**0.0182 ± 0.0016**	**0.0184 ± 0.0009**	**0.0177 ± 0.0021**	**0.0194 ±0.0018**

For these attributes with missing values, we remove all samples with missing values unless the attribute is *OtherPerCap*, whose missing value is replaced by the mean of all the values in this attribute (Chi et al., 2021). After preprocessing, there are 101 attributes left.

Next, we investigate the relationship between the response variable, specifically the *ViolentCrimesPerPop*, and the remaining features. To accomplish this, we employ a simple neural network consisting of three fully-connected layers. The first and second layers of the neural network are fully-connected layers followed by Rectified Linear Unit (ReLU) activation function. The third layer is also a fully-connected layer.

We train it using the Adam optimizer with an initial learning rate of 0.001. The model undergoes training for a total of 50 epochs and set the batchsize equal to 256, with the learning rate decayed by a factor of 10 at the 25th epoch. For the L_2 regularizer, we set the factor to weight_decay = 1e-4. In addition, we fix $\epsilon_1 = 0.017$ for RRonbins, our proposed method and Unbiased across all privacy budgets.

Table 1 provides the quantitative comparison results of the Gaussian, Laplace, Staircase, RRonBins, Unbiased mechanisms and our proposed one for various privacy budgets. Based on the results presented in Table 1, our proposed method is consistently the best across different privacy budgets compared to the other methods. Consequently, we can conclude that our proposed method has good performance compared with the additive noise mechanisms and the RRonBins, Unbiased mechanisms.

5.2 The Criteo Sponsored Search Conversion Log Dataset

The Criteo Sponsored Search Conversion Log Dataset is public and can be downloaded from the Criteo AI Lab website[2]. This dataset represents a sample of 90 days of Criteo live traffic data. Each line corresponds to one click (product related advertisement) that was displayed to a user. For each advertisement, we have detailed information about the product. Further, it also provides information on whether the click led to a conversion, amount of conversion and the time between the click and the conversion. Data has been sub-sampled and anonymized so as not to disclose proprietary elements.

In the Criteo Sponsored Search Conversion Log Dataset, there are 23 attributes (the first three are outcome/responses and the remaining are features). We use 21 of them except *Sale, click_timestamp*, where *SalesAmountInEuro* is considered as the response and the remaining are features. In the features, *Time_delay_for_conversion, nb_clicks_1week* are integers, *product_price* is float, and *product_age_group, device_type, audience_id, product_gender, product_brand, product_category (1-7), product_country, product_id, product_title, partner_id, user_id* are hashed features for user privacy. Therefore, it is necessary to do preprocessing of the hashed features before being used for response DP. We use the very similar technique as that for kaggle-display-Advertising-challenge-dataset (see the webpage `https://github.com/rixwew/pytorch-fm/blob/master/torchfm/dataset/criteo.py`) with a small modification. We count all the unique values for each feature and map them to their corresponding unique value if the occurrence is

[2] `https://ailab.criteo.com/criteo-sponsored-search-conversion-log-dataset/`

large than the preassigned thresholding, otherwise consider as a single out-of-vocabulary. For the architecture of the neural networks used for regression, we benefit from MentorNet (Jiang et al., 2018) used for the non-numerical features (all but attributes *Time_delay_for_conversion, nb_clicks_1week, product_price*) and put them to embeddings. Then we combine the embedded features with *Time_delay_for_conversion, nb_clicks_1week, product_price* together as the input and feed them into a neural network with 3 fully-connected layers.

Now, we are ready for response DP. We plan to predict the relationship between response (*SalesAmountInEuro*) and the features. There are 15,995,634 samples in the dataset, but it is unfortunate that some of them are with no conversion happened (corresponding to *SalesAmountInEuro*$=-1$) and some are too large (the largest one is 62,458.773). We have 1,645,977 samples left after removing those useless ones (-1 or larger than 400). We set the maximum number of epochs to 50 and the batch size to 8192. The model is trained using the RMSProp optimizer with an initial learning rate of 1e-3, and the learning rate is decayed using the *CosineAnnealingLR* method. Additionally, we set the L_2 regularizer parameter weight_decay to 1e-4. In addition, we fix $\epsilon_1 = 0.017$ for RRonbins, our proposed method and Unbiased across all privacy budgets.

Table 2: Comparison results on the Criteo Sponsored Search Conversion Log dataset.

Privacy	Laplace	Gaussian	Staircase	RRonBins	Ours	Unbiased
Budget	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std
0.05	1878491.63 ± 376368.64	16803328.94 ± 998367.25	1736473.22 ± 122674.30	11339.71 ± 36.45	**6903.35 ± 39.18**	10572.59±1134.12
0.1	472390.91 ± 14030.15	4458946.92 ± 385131.83	464807.11 ± 19835.24	11328.04 ± 36.45	**6849.88 ± 60.70**	7884.54 ± 664.82
0.3	60599.02 ± 1124.63	567545.99 ± 8905.76	55470.30 ± 1296.68	11185.20 ± 36.10	6539.40 ± 37.58	**5807.11 ± 393.60**
0.5	24659.75 ± 745.69	215649.49 ± 2694.33	21717.01 ± 417.43	10907.33 ± 36.54	6205.27 ± 84.46	**4826.07 ± 250.91**
0.8	11982.80 ± 290.64	87173.02 ± 1582.45	10328.00 ± 255.31	10256.37 ± 37.39	5900.21 ± 80.57	**4796.94 ± 210.42**
1.0	8854.12 ± 267.56	58036.92 ± 1286.23	7762.13 ± 230.03	9744.08 ± 37.59	5548.51 ± 65.47	**4630.83 ± 171.56**
1.5	5759.41 ± 178.68	28740.18 ± 481.94	5301.81 ± 274.90	8406.88 ± 36.57	4724.96 ± 112.93	**4472.40 ± 92.37**
2	4787.05 ± 127.27	17620.97 ± 493.14	4516.25 ± 62.59	7294.93 ± 34.03	**3970.59 ± 91.30**	4459.39 ± 78.33
3	**3848.77 ±184.59**	9780.89 ± 136.51	4103.87 ± 101.25	5577.50 ± 31.75	**3709.91 ± 236.98**	4330.81 ± 58.60
4	**3551.65 ± 164.53**	7081.23 ± 259.26	4080.65 ± 128.19	4769.61 ± 25.01	**3397.61 ± 116.83**	4244.34 ± 76.74
6	3339.38 ± 141.50	5024.29 ± 340.55	4144.02 ± 75.71	4371.68 ± 25.31	**3109.55 ± 84.33**	3241.17 ± 44.89
8	3164.46 ±83.76	4163.70 ± 119.34	4322.13 ± 187.16	4333.12 ± 31.94	3128.52 ± 88.01	**2911.97 ± 58.3**
$+\infty$	3119.31 ± 99.39	3119.31 ± 99.39	3119.31 ± 99.39	4319.86 ± 29.27	3119.01 ± 70.31	**2798.53±29.11**

In the following, we quantitatively evaluate our proposed method. We compare our method with the Gaussian, Laplace, Staircase, RRonBins, and Unbiased mechanisms. Table 2 presents the quantitative comparison results for the Gaussian, Laplace, and Staircase mechanisms at different privacy budgets.

Based on the results presented in Table 2, our method and the unbiased mechanism are the best two compared with other mechanisms in the Criteo Sponsored Search Conversion Log dataset.

5.3 The California Housing Dataset

This California Housing dataset was derived from the 1990 U.S. census, which can be obtained from the StatLib repository[3] or import through the command *fetch_california_housing*[4] in sklearn.datasets.

In this dataset, we have information regarding the demography (income, population, house occupancy) in the districts, the location of the districts (latitude, longitude), and general information regarding the house in the districts (number of rooms, number of bedrooms, age of the house). There are 20640 instances, 9 attributes: 8 numeric, predictive attributes and the target, where the attribute information is shown in Table 3. Since these statistics are at the granularity of the district, they correspond to averages or medians. There is no missing attribute values, we use the data directly.

For the architecture of the regression neural networks, we employ a 3-layer fully-connected neural network. The model is trained for a total of 50 epochs, with a batch size set to 256. During training, we utilize the Adam optimizer with an initial learning rate of 0.001. The learning rate is decayed by a factor of 10 at the

[3] `https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html`

[4] `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html#sklearn.datasets.fetch_california_housing`

Table 3: Attribute Information.

Name	Description
MedInc	median income in block group
HouseAge	median house age in block group
AveRooms	average number of rooms per household
AveBedrms	average number of bedrooms per household
Population	block group population
AveOccup	average number of household members
Latitude	block group latitude
Longitude	block group longitude

25th epoch. For L_2 regularization, we set the regularization factor to weight_decay = 1e-4. In addition, we fix $\epsilon_1 = 0.007$ for RRonbins, our proposed method and Unbiased across all privacy budgets.

Table 4: Comparison results on the Housing dataset.

Privacy	Laplace	Gaussian	Staircase	RRonBins	Ours	Unbiased
Budget	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std
0.05	14.2061 ± 12.1058	59.1688 ± 25.9966	9.9673 ± 3.7562	**1.3982 ± 0.0365**	**1.4985 ± 0.1574**	8.1585 ± 2.5883
0.1	4.8926 ± 0.8567	14.6442 ± 10.7207	5.6695 ± 1.6705	**1.4147 ± 0.0256**	**1.4656 ± 0.1232**	4.5274 ± 1.1205
0.3	2.2165 ± 0.5216	4.8875 ± 0.9392	2.6639 ± 1.3920	**1.4153 ± 0.0277**	**1.4525 ± 0.1993**	**1.8482 ± 0.2554**
0.5	1.5666 ± 0.2492	3.4958 ± 0.9716	1.4574 ±0.1180	**1.3494 ± 0.0337**	**1.3108 ± 0.0686**	**1.3244 ± 0.1837**
0.8	**1.1604 ±0.1397**	2.8569 ± 0.8632	**1.0189 ± 0.0868**	1.2859 ± 0.0211	**1.1552 ± 0.1247**	**1.0569 ± 0.1152**
1.0	**1.0121 ± 0.1150**	2.3481 ± 0.6479	**0.8862 ±0.0933**	1.2197 ± 0.0472	1.0583 ± 0.0800	**0.8529 ± 0.0724**
1.5	**0.8262 ± 0.0761**	1.6685 ± 0.3689	**0.7905 ± 0.0552**	1.1367 ± 0.1559	**0.8875 ± 0.0792**	**0.8020 ± 0.1106**
2	**0.7527 ± 0.0380**	1.4513 ± 0.2093	**0.7608 ± 0.0559**	0.9697 ± 0.0406	**0.7830 ± 0.0380**	**0.7824 ± 0.1150**
3	0.7279 ± 0.1016	1.0725 ± 0.1550	0.7141 ± 0.0329	0.7706 ± 0.0354	**0.6652 ± 0.0321**	**0.6296 ± 0.0329**
4	**0.6562 ± 0.0495**	0.8827 ± 0.0689	0.7087 ± 0.0573	**0.6572 ± 0.0213**	**0.6378 ± 0.0339**	**0.6199 ± 0.0370**
6	**0.6153 ± 0.0234**	0.7588 ± 0.0725	0.7113 ± 0.0402	**0.6213 ± 0.0146**	**0.6136 ± 0.0356**	**0.6085 ± 0.0157**
8	**0.6065 ± 0.0463**	0.7270 ± 0.0408	0.7166 ± 0.0376	**0.6087 ± 0.0263**	**0.6000 ± 0.0241**	**0.6077 ± 0.0257**
$+\infty$	**0.5922 ± 0.0244**	**0.5922 ± 0.0244**	**0.5922 ± 0.0244**	**0.5912 ± 0.0301**	**0.5852 ± 0.0224**	**0.6110 ±0.0292**

The comparison results can be found in Table 4 on the California Housing dataset. Our method achieves the best performance across all privacy budgets. While the Laplace, Staircase and Unbiased mechanisms perform better on some privacy budgets, our method is consistent in all cases.

6 Conclusions

In this paper, we introduce a novel algorithm for label differential privacy in regression. Our approach, called RPWithPrior, leverages a known global prior distribution to ensure ϵ-label differential privacy. In scenarios where the global prior distribution is unknown, we propose an alternative algorithm that employs a histogram to estimate the prior probability density function. This estimated function is then utilized as a prior in RPWithPrior. Theoretical analysis demonstrates that our proposed algorithms guarantee ϵ-label differential privacy. Furthermore, we conduct numerical experiments to evaluate the effectiveness of our proposed method.

Impact Statement

This paper acknowledges that the numerical implementations are purely illustrative, and are not recommended use cases.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC conference on computer and* *communications security*, pp. 308–318, 2016.

Jayadev Acharya, Kallista Bonawitz, Peter Kairouz, Daniel Ramage, and Ziteng Sun. Context aware local differential privacy. In *International Conference on Machine Learning*, pp. 52–62, 2020.

Ashwinkumar Badanidiyuru, Badih Ghazi, Pritish Kamath, Ravi Kumar, Ethan Jacob Leeman, Pasin Manurangsi, Avinash V. Varadarajan, and Chiyuan Zhang. Optimal unbiased randomizers for regression with label differential privacy. In *Advances in Neural Information Processing Systems*, pp. 60226–60246, 2023.

Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pp. 394–403, 2018.

T.H. Hubert Chan, Elaine Shi, and Dawn Song. Optimal lower bound for differentially private multi-party aggregation. In *European Symposium on Algorithms*, pp. 277–288, 2012.

Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. In *Proceedings of the 24th Annual Conference on Learning Theory*, pp. 155–186. JMLR Workshop and Conference Proceedings, 2011.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3):1069–1109, 2011.

Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 375–403, 2019.

Jianfeng Chi, Yuan Tian, Geoffrey J. Gordon, and Han Zhao. Understanding and mitigating accuracy disparity in regression. In *International Conference on Machine Learning*, pp. 1866–1876, 2021.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual international conference on the theory and applications of cryptographic techniques*, pp. 486–503, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284, 2006b.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2468–2479, 2019.

Hossein Esfandiari, Vahab Mirrokni, Umar Syed, and Sergei Vassilvitskii. Label differential privacy via clustering. In *International Conference on Artificial Intelligence and Statistics*, pp. 7055–7075, 2022.

Natasha Fernandes, Annabelle McIver, and Carroll Morgan. The laplace mechanism has optimal utility for differential privacy over continuous queries. In *The Annual ACM/IEEE Symposium on Logic in Computer Science*, pp. 1–12, 2021.

Quan Geng and Pramod Viswanath. The optimal mechanism in differential privacy. In *IEEE international symposium on information theory*, pp. 2371–2375, 2014.

Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. Deep learning with label differential privacy. *Advances in Neural Information Processing Systems*, 34:27131–27145, 2021.

Badih Ghazi, Pritish Kamath, Ravi Kumar, Ethan Leeman, Pasin Manurangsi, Avinash Varadarajan, and Chiyuan Zhang. Regression with label differential privacy. *arXiv preprint arXiv:2212.06074*, 2022.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2304–2313, 2018.

Mani Malek Esmaeili, Ilya Mironov, Karthik Prasad, Igor Shilov, and Florian Tramer. Antipodes of label differential privacy: Pate and alibi. *Advances in Neural Information Processing Systems*, 34:6934–6945, 2021.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3): 291–297, 1997.

Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

Hai Phan, My T Thai, Han Hu, Ruoming Jin, Tong Sun, and Dejing Dou. Scalable differential privacy with certified robustness in adversarial learning. In *International Conference on Machine Learning*, pp. 7683–7694, 2020.

Michael Redmond and Alok Baveja. A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, 141(3):660–678, 2002.

Abhin Shah, Yuheng Bu, Joshua K. Lee, Subhro Das, Rameswar Panda, Prasanna Sattigeri, and Gregory W. Wornell. Selective regression under fairness criteria. In *International Conference on Machine Learning*, pp. 19598–19615, 2022.

Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the ACM SIGSAC conference on computer and communications security*, pp. 1310–1321, 2015.

Michael T. Smith, Mauricio A. Álvarez, Max Zwiessele, and Neil D. Lawrence. Differentially private regression with gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 1195–1203, 2018.

Michael Thomas Smith, Mauricio A. Álvarez, and Neil D. Lawrence. Differentially private regression and classification with sparse gaussian processes. *Journal of Machine Learning Researh*, 22:1–41, 2021.

Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. Stochastic gradient descent with differentially private updates. In *IEEE global conference on signal and information processing*, pp. 245–248, 2013.

Marcelo Tallis and Pranjul Yadav. Reacting to variations in product demand: An application for conversion rate (cr) prediction in sponsored search. In *IEEE International Conference on Big Data*, pp. 1856–1864, 2018.

Stanley L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: regression analysis under differential privacy. *Proceedings of the VLDB Endowment*, 5(11):1364–1375, 2012.

A Preliminary

Before delving deeper into the topic, it is important to establish the definitions of neighboring datasets, differential privacy and label differential privacy.

Definition A.1 (Neighboring datasets (Dwork et al., 2006b))**.** *Two datasets D and D' are called the neighboring datasets if they differ in at most one elements. That is, one is a proper subset of the other and the larger dataset contains just one additional element.*

Definition A.2 (Differential Privacy (Dwork et al., 2006b))**.** *Let $\mathcal{A}$ be a randomized algorithm which takes a dataset as an input. For any $\epsilon, \delta \in \mathbb{R}_+$, $\mathcal{A}$ is said to be (ϵ, δ)-differentially private ((ϵ, δ)-DP) if $P[\mathcal{A}(\mathcal{D}) \in \mathcal{S}] \leq e^{\epsilon} P[\mathcal{A}(\mathcal{D}') \in \mathcal{S}] + \delta$ holds for any neighboring training sets $\mathcal{D}, \mathcal{D}'$ and any output sets $\mathcal{S}$ of $\mathcal{A}$. When $\delta = 0$, we call $\mathcal{A}$ is ϵ-differential privacy (ϵ-DP).*

The definition of label differential privacy can be stated as follows:

Definition A.3 (Label Differential Privacy (Chaudhuri & Hsu, 2011))**.** *Let $\mathcal{A}$ be a randomized algorithm that takes a dataset as an input. For any $\epsilon, \delta \in \mathbb{R}_+$, $\mathcal{A}$ is said to be (ϵ, δ)-label differentially private ((ϵ, δ)-label DP) if the following condition holds for any training sets $\mathcal{D} = [\mathcal{X}, \mathcal{Y}], \mathcal{D}' = [\mathcal{X}, \mathcal{Y}']$ that differ only in the label of a single example and any output sets $\mathcal{S}$ of $\mathcal{A}$: $P[\mathcal{A}(\mathcal{Y}) \in \mathcal{S}] \leq e^{\epsilon} P[\mathcal{A}(\mathcal{Y}') \in \mathcal{S}] + \delta$. When $\delta = 0$, we refer to $\mathcal{A}$ as ϵ-label differential private (ϵ-label DP).*

B Proof of Lemma 3.3

Proposition B.1. *Let $\mathcal{I} = [A_1, A_2]$ and $\gamma = 2\zeta + e^{-\epsilon}(A_2 - A_1)$. Assume $f_{\tilde{Y}|Y}(\tilde{y}, y) = v \geq 0, \forall y \in \mathcal{I}, \forall \tilde{y} \in \mathcal{N}_y$. When $y \in \mathcal{I}$, we can make the following statement*

$$\int_{\mathcal{N}_{\mathcal{I}}} f_{\tilde{Y}|Y}(\tilde{y}, y) d\tilde{y} \geq \gamma \cdot v. \tag{B.1}$$

Proof. For any $y \in \mathcal{I}$, we have

$$\begin{aligned}
\int_{\mathcal{N}_{\mathcal{I}}} f_{\tilde{Y}|Y}(\tilde{y}, y) d\tilde{y} &= \int_{\mathcal{N}_y} f_{\tilde{Y}|Y}(\tilde{y}, y) d\tilde{y} + \sum_{k \neq 0} \int_{\mathcal{N}_{y+2k\zeta} \cap \mathcal{N}_{\mathcal{I}}} f_{\tilde{Y}|Y}(\tilde{y}, y) d\tilde{y} \\
&\geq \int_{\mathcal{N}_y} f_{\tilde{Y}|Y}(\tilde{y}, y) d\tilde{y} + e^{-\epsilon} \sum_{k \neq 0} \int_{\mathcal{N}_{y+2k\zeta} \cap \mathcal{N}_{\mathcal{I}}} f_{\tilde{Y}|Y}(\tilde{y}, P_{\mathcal{I}}(y + 2k\zeta)) d\tilde{y},
\end{aligned}$$

where we have used the fact that $e^{\epsilon} f_{\tilde{Y}|Y}(\tilde{y}, y) \geq f_{\tilde{Y}|Y}(\tilde{y}, P_{\mathcal{I}}(y + 2k\zeta))$ for all $\tilde{y} \in \mathcal{N}_{y+2k\zeta} \cap \mathcal{N}_{\mathcal{I}}$ and

$$P_{\mathcal{I}}(x) = \begin{cases} A_2, & x > A_2, \\ x, & x \in \mathcal{I}, \\ A_1, & x < A_1. \end{cases}$$

- When $y + 2k\zeta \in \mathcal{I}$ and $\tilde{y} \in \mathcal{N}_{y+2k\zeta} \cap \mathcal{N}_{\mathcal{I}}$, we have $P_{\mathcal{I}}(y + 2k\zeta) = y + 2k\zeta$, $\tilde{y} \in \mathcal{N}_{y+2k\zeta}$ and $|\tilde{y} - P_{\mathcal{I}}(y + 2k\zeta)| \leq \zeta$,
- when $y + 2k\zeta > A_2$ and $\tilde{y} \in \mathcal{N}_{y+2k\zeta} \cap \mathcal{N}_{\mathcal{I}}$, we have $y + (2k-1)\zeta \leq \tilde{y} \leq A_2 + \zeta$, $P_{\mathcal{I}}(y + 2k\zeta) = A_2$ and

$$\begin{aligned}
\tilde{y} - P_{\mathcal{I}}(y + 2k\zeta) =& \tilde{y} - A_2 \geq \tilde{y} - (y + 2k\zeta) \geq (y + (2k-1)\zeta) - (y + 2k\zeta) = -\zeta, \\
\tilde{y} - P_{\mathcal{I}}(y + 2k\zeta) =& \tilde{y} - A_2 \leq A_2 + \zeta - A_2 = \zeta,
\end{aligned}$$

- when $y + 2k\zeta < A_1$ and $\tilde{y} \in \mathcal{N}_{y+2k\zeta} \cap \mathcal{N}_{\mathcal{I}}$, we have $A_1 - \zeta \leq \tilde{y} \leq y + (2k+1)\zeta$, $P_{\mathcal{I}}(y + 2k\zeta) = A_1$ and

$$\begin{aligned}
\tilde{y} - P_{\mathcal{I}}(y + 2k\zeta) =& \tilde{y} - A_1 \leq \tilde{y} - (y + 2k\zeta) \leq (y + (2k+1)\zeta) - (y + 2k\zeta) = \zeta, \\
\tilde{y} - P_{\mathcal{I}}(y + 2k\zeta) =& \tilde{y} - A_1 \geq A_1 - \zeta - A_1 = -\zeta,
\end{aligned}$$

which implies $\tilde{y} \in \mathcal{N}_{P_{\mathcal{I}}(y+2k\zeta)}$ when $\tilde{y} \in \mathcal{N}_{y+2k\zeta} \cap \mathcal{N}_{\mathcal{I}}$. Note that $P_{\mathcal{I}}(y + 2k\zeta) \in \mathcal{I}$, we have $f_{\tilde{Y}|Y}(\tilde{y}, P_{\mathcal{I}}(y + 2k\zeta)) = v$, from the assumption $f_{\tilde{Y}|Y}(\tilde{y}, y) = v \geq 0, \forall y \in \mathcal{I}, \forall \tilde{y} \in \mathcal{N}_y$. Then

$$\int_{\mathcal{N}_{\mathcal{I}}} f_{\tilde{Y}|Y}(\tilde{y}, y) d\tilde{y} \geq [2\zeta + e^{-\epsilon}(A_2 - A_1)] \cdot v.$$

□

Proof of Lemma 3.3. When $y \in \mathcal{I}$, Equation (B.1) holds from Proposition B.1. We can conclude that the first and second constraints of equation (3.3) imply that

$$1 \geq \int_{\mathcal{N}_{\mathcal{I}}} f_{\tilde{Y}|Y}(\tilde{y}, y) d\tilde{y} \geq \gamma \cdot v, \ \forall y \in \mathcal{I}.$$

which implies $v \leq 1/\gamma$. Then

$$\int_{\mathcal{I}} f_Y(y) \left[\int_{\mathcal{N}_y} f_{\tilde{Y}|Y}(\tilde{y}, y) d\tilde{y} \right] dy = 2\zeta v \int_{\mathcal{I}} f_Y(y) dy \leq \frac{2\zeta}{\gamma} \int_{\mathcal{I}} f_Y(y) dy.$$

When $v = 1/\gamma$, the objective function achieve the maximal value. By the second constraint in (3.3), we have

$$f_{\tilde{Y}|Y}(\tilde{y}, y') \geq \frac{e^{-\epsilon}}{\gamma}, \forall \tilde{y} \in \mathcal{N}_{\mathcal{I}}, y' \in \mathcal{I}. \tag{B.2}$$

From (B.2) and $f_{\tilde{Y}|Y}(\tilde{y}, y') = 1/\gamma, \forall \tilde{y} \in \mathcal{N}_{y'}, y' \in \mathcal{I}$ when the objective function achieve the maximal value, we have

$$\frac{e^{-\epsilon}(A_2 - A_1)}{\gamma} \leq \int_{\mathcal{N}_{\mathcal{I}}/\mathcal{N}_{y'}} f_{\tilde{Y}|Y}(\tilde{y}, y') d\tilde{y} \leq 1 - \int_{\mathcal{N}_{y'}} f_{\tilde{Y}|Y}(\tilde{y}, y') d\tilde{y} = \frac{e^{-\epsilon}(A_2 - A_1)}{\gamma},$$

which implies

$$f_{\tilde{Y}|Y}(\tilde{y}, y') = \begin{cases} \frac{e^{-\epsilon}}{\gamma}, & \forall \tilde{y} \in \mathcal{N}_{\mathcal{I}}/\mathcal{N}_{y'}, y' \in \mathcal{I}, \\ 0, & \forall \tilde{y} \notin \mathcal{N}_{\mathcal{I}}. \end{cases} \text{ almost everywhere.}$$

We conclude the result. □

C Proof of Lemma 3.5

Proof. We get the derivatives about A_1, A_2, then

$$\begin{aligned}
&\frac{\partial F(A_1, A_2)}{\partial A_1} \\
=&[-f_Y(A_1)] \frac{2\zeta}{2\zeta + e^{-\epsilon}(A_2 - A_1)} + \int_{A_1}^{A_2} f_Y(y) dy (2\zeta)(-1)(2\zeta + e^{-\epsilon}(A_2 - A_1))^{-2}(-1)e^{-\epsilon} \\
=&\frac{-2\zeta}{[2\zeta + e^{-\epsilon}(A_2 - A_1)]^2} \left[f_Y(A_1)(2\zeta + e^{-\epsilon}(A_2 - A_1)) - e^{-\epsilon} \int_{A_1}^{A_2} f_Y(y) dy \right] \\
\triangleq&\frac{-2\zeta}{[2\zeta + e^{-\epsilon}(A_2 - A_1)]^2} g_1(A_1, A_2),
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial F(A_1, A_2)}{\partial A_2} =&\frac{2\zeta}{[2\zeta + e^{-\epsilon}(A_2 - A_1)]^2} \left[f_Y(A_2)(2\zeta + e^{-\epsilon}(A_2 - A_1)) - e^{-\epsilon} \int_{A_1}^{A_2} f_Y(y) dy \right] \\
\triangleq&\frac{2\zeta}{[2\zeta + e^{-\epsilon}(A_2 - A_1)]^2} g_2(A_1, A_2),
\end{aligned}$$

where

$$\begin{aligned}
g_1(A_1, A_2) &= f_Y(A_1)(2\zeta + e^{-\epsilon}(A_2 - A_1)) - e^{-\epsilon} \int_{A_1}^{A_2} f_Y(y) dy, \\
g_2(A_1, A_2) &= f_Y(A_2)(2\zeta + e^{-\epsilon}(A_2 - A_1)) - e^{-\epsilon} \int_{A_1}^{A_2} f_Y(y) dy.
\end{aligned}$$

When $A_1 < n_0$, $\frac{\partial F(A_1,A_2)}{\partial A_1} = \frac{2\zeta}{[2\zeta + e^{-\epsilon}(A_2 - A_1)]^2} e^{-\epsilon} \int_{A_1}^{A_2} f_Y(y) dy > 0$ for any $A_2 > A_1$. Then $F(A_1, A_2)$ is increasing when $A_1 \in (-\infty, n_0)$. Similarly, When $A_2 > n_k$, $\frac{\partial F(A_1,A_2)}{\partial A_2} = -\frac{2\zeta}{[2\zeta + e^{-\epsilon}(A_2 - A_1)]^2} e^{-\epsilon} \int_{A_1}^{A_2} f_Y(y) dy < 0$ for any $A_1 < A_2$. Then $F(A_1, A_2)$ is decreasing when $A_2 \in (n_k, \infty)$.

In the following, we consider $A_1, A_2 \in [n_0, n_k]$. Assume $A_1 \in [n_i, n_{i+1})$ and $A_2 \in [n_j, n_{j+1}]$ with $i \leq j$, then

$$g_1(A_1, A_2) = \alpha_i(2\zeta + (A_2 - A_1)e^{-\epsilon}) - e^{-\epsilon} \left[\alpha_i(n_{i+1} - A_1) + \sum_{\ell=i+1}^{j-1} \alpha_\ell(n_{\ell+1} - n_\ell) + \alpha_j(A_2 - n_j) \right]$$

$$=e^{-\epsilon}(\alpha_i-\alpha_j)A_2+c_1=-(e^{-\epsilon}(\alpha_j-\alpha_i)A_2-c_1),$$

$$\begin{aligned}g_2(A_1,A_2)=&\alpha_j(2\zeta+(A_2-A_1)e^{-\epsilon})-e^{-\epsilon}\left[\alpha_i(n_{i+1}-A_1)+\sum_{\ell=i+1}^{j-1}\alpha_\ell(n_{\ell+1}-n_\ell)+\alpha_j(A_2-n_j)\right]\\=&e^{-\epsilon}(\alpha_i-\alpha_j)A_1+c_2,\end{aligned}$$

where $c_1=2\zeta\alpha_i-e^{-\epsilon}h$, $c_2=2\zeta\alpha_j-e^{-\epsilon}h$ and $h=\alpha_i n_{i+1}+\sum_{\ell=i+1}^{j-1}\alpha_\ell(n_{\ell+1}-n_\ell)-\alpha_j n_j$.

We also discuss the monotonicity of $F(A_1,A_2)$ as follows:

Define $d_{11}=e^{-\epsilon}(\alpha_j-\alpha_i)n_j-c_1$, $d_{12}=e^{-\epsilon}(\alpha_j-\alpha_i)n_{j+1}-c_1$.

- when $\max(d_{11},d_{12})<0$, $F(A_1,A_2)$ is decreasing about $A_1\in(n_i,n_{i+1})$ when $A_2\in(n_j,n_{j+1})$,
- when $\min(d_{11},d_{12})>0$, $F(A_1,A_2)$ is increasing about $A_1\in(n_i,n_{i+1})$ when $A_2\in(n_j,n_{j+1})$,
- when $d_{11}d_{12}<0$, let $e_2=\frac{c_1}{e^{-\epsilon}(\alpha_j-\alpha_i)}$, which must be in the interval (n_j,n_{j+1}).
  - When $d_{11}<0$ and $d_{12}>0$, $F(A_1,A_2)$ is decreasing about $A_1\in(n_i,n_{i+1})$ for $A_2\in(n_j,e_2)$ and $F(A_1,A_2)$ is increasing about $A_1\in(n_i,n_{i+1})$ for $A_2\in(e_2,n_{j+1})$.
  - When $d_{11}>0$ and $d_{12}<0$, $F(A_1,A_2)$ is increasing about $A_1\in(n_i,n_{i+1})$ for $A_2\in(n_j,e_2)$ and $F(A_1,A_2)$ is decreasing about $A_1\in(n_i,n_{i+1})$ for $A_2\in(e_2,n_{j+1})$.

Define $d_{21}=e^{-\epsilon}(\alpha_i-\alpha_j)n_i+c_2$ and $d_{22}=e^{-\epsilon}(\alpha_i-\alpha_j)n_{i+1}+c_2$. Therefore,

- when $\max(d_{21},d_{22})<0$, $F(A_1,A_2)$ is decreasing about $A_2\in(n_j,n_{j+1})$ when $A_1\in(n_i,n_{i+1})$,
- when $\min(d_{21},d_{22})>0$, $F(A_1,A_2)$ is increasing about $A_2\in(n_j,n_{j+1})$ when $A_1\in(n_i,n_{i+1})$,
- when $d_{21}d_{22}<0$, let $e_1=\frac{c_2}{e^{-\epsilon}(\alpha_j-\alpha_i)}$, which must be in the interval (n_i,n_{i+1}).
  - When $d_{21}<0$ and $d_{22}>0$, $F(A_1,A_2)$ is decreasing about $A_2\in(n_j,n_{j+1})$ for $A_1\in(n_i,e_1)$ and $F(A_1,A_2)$ is increasing about $A_2\in(n_j,n_{j+1})$ for $A_1\in(e_1,n_{i+1})$.
  - When $d_{21}>0$ and $d_{22}<0$, $F(A_1,A_2)$ is increasing about $A_2\in(n_j,n_{j+1})$ for $A_1\in(n_i,e_1)$ and $F(A_1,A_2)$ is decreasing about $A_2\in(n_j,n_{j+1})$ for $A_1\in(e_1,n_{i+1})$.

□

D Details of Numerical Experiments

D.1 Parameters

We choose parameters manually, similar as grid search. In addition, we consider σ as a parameter. In empirical tests, varying $\sigma=\alpha\sigma_1$ by a factor of α, where σ_1 is the standard deviation. In the following, we will report the parameters used in the Communities and Crime Dataset, the Criteo Sponsored Search Conversion Log Dataset and the California Housing Dataset.

Table 5: Parameters.

Crime			Criteo			Housing		
ϵ	α	ζ	ϵ	α	ζ	ϵ	α	ζ
0.05	2	0.2	0.05	0.8	100	0.05	2	0.7
0.1	2	0.2	0.1	0.8	100	0.1	2	0.6
0.3	2	0.2	0.3	0.5	100	0.3	2	1.2
0.5	2	0.2	0.5	0.5	100	0.5	1.5	1.4
0.8	1.5	0.2	0.8	0.08	50	0.8	1.5	1.5
1	1.5	0.2	1	0.08	80	1	1.5	1.5
1.5	1.5	0.2	1.5	0.08	80	1.5	1.5	1.5
2	1.5	0.3	2	0.08	100	2	1.5	1.5
3	1.5	0.3	3	0.2	60	3	1.5	1.4
4	1.5	0.3	4	0.2	60	4	0.4	0.25
6	0.03	0.3	6	0.2	30	6	0.03	0.2
8	0.03	0.3	8	0.08	20	8	0.025	0.1
$+\infty$	$-$	0.1	$+\infty$	$-$	0.8	$+\infty$	$-$	0.1

D.2 Computational Complexity

To save storage space and reduce computational costs, the authors of RRonBins discretized the labels by rounding down the original responses to integer values (referred to as discretized samples). For the remaining methods, the original responses are retained.

Table 6: Runtime and Unique number (n) of samples or discretized samples in the training process.

Method	Crime		Criteo		Housing	
	Time	n	Time	n	Time	n
Laplace	0.288116455	1595	**3.408373356**	1316781	**0.279757977**	16512
Gaussian	0.285983801	1595	3.42813	1316781	0.291029215	16512
Staircase	0.285481691	1595	3.538956881	1316781	0.312218666	16512
RRonBins	0.567934513	94	8.24254632	401	6.187591076	472
Ours	**0.279293299**	1595	3.44035	1316781	0.297134399	16512

In Table 6, we provide the runtime and unique number of samples in the training process. Our method is computationally efficient, comparable to additive noise mechanisms and significantly outperforming the RRonBins mechanism.