Smoothly Bounding User Contributions in Differential Privacy

Alessandro Epasto
Google Research
111 8th Ave,
New York, NY, 10011
aepasto@google.com

Mohammad Mahdian
Google Research
111 8th Ave,
New York, NY, 10011
mahdian@google.com

Jieming Mao
Google Research
111 8th Ave,
New York, NY, 10011
maojm@google.com

Vahab Mirrokni
Google Research
111 8th Ave,
New York, NY, 10011
mirrokni@google.com

Lijie Ren
Google Research
111 8th Ave,
New York, NY, 10011
renlijie@google.com

Abstract

A differentially private algorithm guarantees that the input of a single user won't significantly change the output distribution of the algorithm. When a user contributes more data points, more information can be collected to improve the algorithm's performance. But at the same time, more noise might need to be added to the algorithm in order to keep the algorithm differentially private and this might hurt the algorithm's performance. [AKMV19] initiates the study on bounding user contributions and proposes a very natural algorithm which limits the number of samples each user can contribute by a threshold.

For a better trade-off between utility and privacy guarantee, we propose a method which smoothly bounds user contributions by setting appropriate weights on data points and apply it to estimating the mean/quantiles, linear regression, and empirical risk minimization. We show that our algorithm provably outperforms the sample limiting algorithm. We conclude with experimental evaluations which validate our theoretical results.

1 Introduction

The notion of *Differential Privacy*, introduced by [DMNS06], aims to capture the requirement that the output of an algorithm should not reveal much about the information provided by a single user. The classical definition of differential privacy assumes each user controls one row in the input data set, and guarantees that the removal (or change) of one row in the data set does not change the output significantly.

In many applications of differential privacy, a single user might contribute more than one data point. A prominent example, which is the focus of this paper, is private machine learning, where a user often provides several points in the training data set. While the standard definition of differential privacy can still capture such settings by defining a row as the collection of all data points belonging to the same user, an important and useful nuance is lost in this translation. Most importantly, when a user contributes many data points, the algorithm designer must balance between the value of the information contained in these data points, and the added noise it will have to add to the output to make it private with respect to this user.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

[AKMV19] initiated the study of this problem, and proposed a natural algorithm which limits the number of samples each user can contribute by a threshold. This threshold is then optimized to strike the right balance between the error due to the noise, and the bias introduced by removing the samples.

This sample limiting algorithm has two drawbacks: (i) It completely discards some data points from users who have too many data points and the information of these data points is lost. (ii) Some data points may contain more useful information than the others but the sample limiting algorithm treats all data points the same when deciding which data points to discard. Our goal in this paper is to answer this question: is it possible to significantly improve over sample limiting by bounding the contribution of each user in a way that is more smooth and careful about the information contained in each sample?

To answer this question, we propose a weighted averaging method to smoothly bound user contributions. The main idea of this method is to set appropriate weights on data points instead of completely discarding some data points.

1.1 Our results

In Section 3, as a warm-up, we study a simple problem: estimating the mean. For this problem, finding the optimal algorithm corresponds to finding the right weights when averaging samples. We compute the overall error of the algorithm in terms of these weights, and show how the optimal set of weights can be found. We then compare the error of such an optimal algorithm with that of the best sample limiting approach. We present examples showing that the error of the sample limiting method can be asymptotically 1.5 times higher than that of our algorithm. However, as we prove, this gap cannot exceed 4.

In Section 4, we extend the weighted averaging algorithm to empirical risk minimization by minimizing a weighted version of empirical risk. Our main technical contribution is to prove a weighted version of uniform convergence, which could be of independent interest. We also extend the weighted averaging algorithm to estimating quantiles in a similar way (in Appendix B). Similarly as the warm-up problem, for ERM and estimating quantiles, our weighted averaging algorithm has advantage over the sample limiting algorithm, but the advantage is limited.

In Section 5, we study linear regression with label privacy (defined in Section 5). We show that label privacy allows us to design the weights better based on the usefulness of data points and the weighted averaging algorithm can have a much bigger advantage over the sample limiting algorithm. In particular, we prove that the optimal algorithm can be parameterized by a matrix C, and calculate the error as a function of this matrix C. Next, we prove that this function is convex, and therefore, one can compute the optimal C in polynomial time. In Section 5.3, we study the gap between the error of our algorithm and the sample limiting algorithm with the best possible threshold, and prove that this gap can be unbounded. In other words, there are instances where our algorithm has an error that is better than the best sample limiting algorithm by an arbitrary factor.

Finally, in Section 6, we analyze the performance of our algorithm and its comparison with sample limiting empirically using some real-world data sets as well as generated data for linear regression with label privacy. This empirical study shows that our algorithms achieve lower loss compared to the baseline methods (e.g., sample limiting) – confirming our theoretical results. We also include in Appendix E experiment results on logistic regression using our ERM algorithm.

1.2 Related work

Differential privacy is proposed by the seminal work of [DMNS06]. For a detailed survey on differential privacy, see [DR14].

Differentially private linear regression and its general form, empirical risk minimization have been well-studied [CM08, CMS11, KST12, JKT12, TS13, SCS13, DJW13, JT14, BST14, Ull15, TTZ15, STU17, WLK$^+$17, WYX17, ZZMW17, Wan18, She19a, She19b, INS$^+$19, BFTT19, WX19, FKT20]. In particular, [WX19] studies label privacy which is similar to the setting we have in Section 5. These results are in the case when each user has only one data point.

Motivated by federated learning, [AKMV19] initiates the study of bounding user contributions in differential privacy. [TAM19, PSY$^+$19] study how to adaptively bound user contributions in

differentially private stochastic gradient descent for federated learning. For a detailed survey on federated learning, see [KMA+19]. More broadly, our setting of each user having multiple data points can be considered as a special case of personalized/heterogeneous differential privacy [JYC15, SCS15, AGK17] and is very related to group privacy which is introduced in [Dwo06].

2 Preliminaries

2.1 Differential privacy

We define the notion of *differential privacy* for an algorithm $\mathcal{A}$ that takes as input a data set D from the space $\mathcal{D}$ of all possible data sets, and produces an output $\mathcal{A}(D)$ in the space of outputs $\mathcal{O}$. Typically, D is a collection of n data points for some n. To define differential privacy, we need a notion of *neighboring* data sets. In the classical setting of differential privacy, two data sets $D, D' \in \mathcal{D}$ are called neighboring data sets, denoted $D \sim D'$, if one is obtained from the other by removing one data point. In Section 2.2, we will discuss a more general notion that captures settings where a user controls more than one data point.

Definition 1 (Differential Privacy [DMNS06])**.** *A randomized algorithm $\mathcal{A}$ is (ε, δ)-differentially private ((ε, δ)-DP for short) if for all neighboring data sets $D, D' \in \mathcal{D}$, and all subsets of outcomes $S \subseteq \mathcal{O}$,*

$$\Pr[\mathcal{A}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{A}(D') \in S] + \delta.$$

When $\delta = 0$, we say that $\mathcal{A}$ is ε-DP.

The Laplace mechanism [DMNS06] is a standard technique to achieve differential privacy by adding Laplace noise of appropriate scale to the outcome of computation.

Definition 2 (ℓ_1-sensitivity)**.** *The ℓ_1-sensitivity of a function $f : \mathcal{D} \to \mathbb{R}^d$ is: $\Delta f = \max_{D \sim D'} \|f(D) - f(D')\|_1$.*

Definition 3 (Laplace Mechanism [DMNS06])**.** *Given any function $f : \mathcal{D} \to \mathbb{R}^d$, the Laplace mechanism is defined as $f(D) + (W_1, ..., W_d)$ where W_i's are i.i.d random variables drawn from Lap$(\Delta f/\varepsilon)$. Here Lap(b) is the Laplace distribution with mean 0 and variance $2b^2$.*

Theorem 1 ([DMNS06])**.** *The Laplace mechanism is ε-DP.*

2.2 User-level differential privacy

In this paper, we consider the setting in which there are m users owning n data points and $m \leq n$. Therefore, a single user can have more than one data point. For each user $l \in [m]$, we use S_l to denote the set of indices of data points owned by user l, and let $s_l = |S_l|$. We assume S_l's are publicly known. We focus on the case when user data points are sampled from the same distribution. When user data distributions are heterogeneous, additional bias need to be deal with (as in [AKMV19]), and we do not consider this case.

The user-level differential privacy definition mostly follows Definition 1. The only difference is that now two data sets D, D' are considered to be neighboring data sets if they are the same except all data points from a single user l.

3 Warm-up: estimating the mean

For warm-up, we consider a simple setting where we have n data points $y_1, ..., y_n$ generated as $y_i = \beta + \xi_i$ and we want to estimate the unknown mean β differentially privately. Here noise ξ_i's are independent with mean 0 and variance σ^2. We assume all y_i's are bounded, i.e., $y_i \in [0, B]$.

We want to minimize the expected squared error of our estimate $\tilde{\beta}$: $\mathbb{E}[(\beta - \tilde{\beta})^2]$. This is just the variance of $\tilde{\beta}$ when β is unbiased ($\mathbb{E}[\tilde{\beta}] = \beta$). The expectation is over the randomness of our algorithm and ξ_i's.

3.1 The weighted averaging algorithm

In this subsection, we propose our weighted averaging algorithm WA_c which is parameterized by non-negative weights $c_1, ..., c_n$ with $\sum_i c_i = 1$. This algorithm simply computes the weighted average of the input y_i's, and applies the Laplace mechanism to this average:

Algorithm 1 Weighted Averaging WA_c

Input: $y_1, \ldots, y_n \in \mathbb{R}$

Parameters: $c_1, \ldots, c_n \in \mathbb{R}_{\geq 0}$, with $\sum_{i=1}^n c_i = 1$

1: $\hat{\beta} \leftarrow c_1 y_1 + \cdots + c_n y_n$

2: $\tilde{\beta} \leftarrow \hat{\beta} + \text{Lap}\left(\frac{B \cdot \max_{l=1}^m \sum_{i \in S_l} c_i}{\varepsilon}\right)$

3: **return** $\tilde{\beta}$

In the following two lemmas, we prove that WA_c is ε-DP and analyze its variance.

Lemma 1. *For every c, the algorithm WA_c is ε-DP.*

Proof. If some user l change its input y_i's for $i \in S_l$, $\hat{\beta}$ would be changed additively by at most $B \cdot \sum_{i \in S_l} c_i$. Therefore, the ℓ_1-sensitivity of $\hat{\beta}$ is $B \cdot \max_l \sum_{i \in S_l} c_i$. By Theorem 1, we know that $\tilde{\beta}$ is ε-DP. □

It is easy to check that $\tilde{\beta}$ is unbiased. We will just analyze its variance.

Lemma 2. *For every c, the variance of the output of WA_c can be written as: Var $\left(\tilde{\beta}\right) = \sigma^2(c_1^2 + \cdots + c_n^2) + 2\left(\frac{B \cdot \max_{l=1}^m \sum_{i \in S_l} c_i}{\varepsilon}\right)^2$.*

Proof. For the variance of $\tilde{\beta}$, since $c_1 y_1, ..., c_n y_n$ and the Laplace noise are independent, we have

$$\begin{aligned}\text{Var}\left(\tilde{\beta}\right) &= \sum_{i=1}^n \text{Var}(c_i y_i) + \text{Var}\left(\text{Lap}\left(\frac{B \cdot \max_{l=1}^m \sum_{i \in S_l} c_i}{\varepsilon}\right)\right) \\ &= \sigma^2(c_1^2 + \cdots + c_n^2) + 2\left(\frac{B \cdot \max_{l=1}^m \sum_{i \in S_l} c_i}{\varepsilon}\right)^2.\end{aligned}$$

□

Next, we characterize the weight vector c that minimizes Var$(\tilde{\beta})$. The proof of Lemma 3 can be found in the supplementary material.

Lemma 3. *Let $c^* = (c_1^*, ..., c_n^*)$ be the vector that minimizes Var$(\tilde{\beta})$. There exists h, such that,*

1. *$s_1 \leq h \leq s_m$*
2. *For each data point i of user q, $c_i^* = \frac{\min(h, s_q)}{s_q \sum_{l=1}^m \min(h, s_l)}$.*

Define $n_h = \sum_{l=1}^m \min(h, s_l)$. Using the characterization we get in Lemma 3, we show in the next claim that minimizing Var$(\tilde{\beta})$ can be simplified into minimizing a function of a single variable h.

Claim 1. *For the weighted averaging algorithm, the minimum of Var$(\tilde{\beta})$ equals to $\min_{h: s_1 \leq h \leq s_m} v(h)$, where $v(h) = \sigma^2 \sum_{l=1}^m s_l \cdot \left(\frac{\min(h, s_l)}{n_h \cdot s_l}\right)^2 + 2\left(\frac{B \cdot h}{\varepsilon \cdot n_h}\right)^2$.*

Proof. For any $h > 0$, when setting $c_i = \frac{\min(h, s_q)}{s_q \sum_{l=1}^m \min(h, s_l)}$ for any data point i of any user q, it is easy to check that Var$(\tilde{\beta}) = v(h)$. Then by Lemma 3, we get the claim. □

Regarding weights computation, Lemma 3 and Claim 1 show that finding the optimal weight vector c is equivalent to minimizing a function $v(h)$ of a single parameter $h \in [s_1, s_m]$. Optimizing this function of a single parameter can be simply done by setting the derivative to be 0 and considering locations where the derivative is not continuous.

3.2 The sample limiting algorithm

The sample limiting algorithm picks an integer threshold h between s_1 and s_m. For each user l, the sample limiting algorithm arbitrarily selects $\min(s_l, h)$ data points and apply the Laplace mechanism to the average of $n_h = \sum_{l=1}^{m} \min(h, s_l)$ selected data points. In other words, if we let T denote the set of selected samples, the sample limiting algorithm outputs $\tilde{\beta} = \frac{1}{n_h} \sum_{i \in T} y_i + \text{Lap}\left(\frac{B \cdot h}{\varepsilon \cdot n_h}\right)$.

It is easy to see that the sample limiting algorithm is a special case of the weighted averaging algorithm with weights $c_i = \mathbb{1}[i \in T]/n_h$. Therefore, by Lemma 1, we know the output of the sample limiting algorithm is ε-DP. By Lemma 2, the variance of $\tilde{\beta}$ can be written as follows.

Claim 2. *For the sample limiting algorithm with integer threshold h, Var$(\tilde{\beta})$ can be written as the following function: $v'(h) = \frac{\sigma^2}{n_h} + 2\left(\frac{B \cdot h}{\varepsilon \cdot n_h}\right)^2$.*

3.3 Comparing the variances

In this subsection, we compare the minimum variances of two algorithms we describe earlier. By Claims 1 and 2, we just need to compare $\min_{h: s_1 \le h \le s_m} v(h)$ and $\min_{h: s_1 \le h \le s_m, h \in \mathbb{N}} v'(h)$.

Since the sample limiting algorithm is a special case of the weighted averaging algorithm, we know its variance is greater than or equal to that of the best weighted averaging algorithm. We now show that there are examples where the variance of the sample limiting algorithm is larger than the variance of the weighted averaging algorithm by a factor that asymptotically converges to $3/2$.

Theorem 2. *For every $g \in \mathbb{N}$, there is an instance where the variance of the best weighted averaging algorithm is less than $\frac{2g+1}{4g^2}$, while the variance of the best sample limiting algorithm is at least $\frac{3}{4g}$. For every g, $\frac{3}{4g} \ge \frac{2g+1}{4g^2}$, and the ratio between these two numbers converges to $3/2$ as g goes to infinity.*

On the other hand, we show that the gap between the two minimum variances is at most a factor of 4.

Theorem 3. *In every instance, the variance of the best sample limiting algorithm is at most 4 times the variance of the best weighted averaging algorithm.*

4 Extension to empirical risk minimization

In this section, we extend the weighted averaging algorithm to empirical risk minimization. Missing proofs and similar extension to estimating quantiles can be found in the supplementary material.

Here we give the setting of empirical risk minimization (ERM). We are given n data points $D = (X_1, ..., X_n)$ from a universe $\mathcal{X}$. They are sampled independently from an unknown distribution μ. We need to optimize over a closed, convex set $\mathcal{C}$ bounded by B (i.e. for all $\theta \in \mathcal{C}$, $\|\theta\|_2 \le B$) and we are given a loss function l. For each data point $X \in \mathcal{X}$, $l(\cdot, X)$ defines a loss function on $\mathcal{C}$. We assume $l(\cdot, X)$ is convex and L-Lipschitz. Our goal is to minimize the population risk $L_\mu(\theta) = \mathbb{E}_{X \sim \mu}[l(\theta, X)]$ over $\theta \in \mathcal{C}$ and we define $\theta^* \in \mathcal{C}$ to be the optimal solution: $\theta^* \in \arg\min_{\theta \in \mathcal{C}} L_\mu(\theta)$.

Now we describe our weighted ERM algorithm (Algorithm 2) parametrized by non-negative weights $c = (c_1, ..., c_n)$. The main idea is to consider the weighted empirical risk $\hat{L}(\theta, c, D) = \sum_{i=1}^{n} c_i l(\theta, X_i)$ for dataset $D = (X_1, ..., X_n)$. In order to apply standard (record-level) differentially private ERM algorithms and ensure differential privacy with respect to each user, we define a new loss function l'. Let M_j to be the meta-data of user j: $M_j = (S_j, \{c_i, X_i\}_{i \in S_j})$. We define $l'(\theta, M_j) = \sum_{i \in S_j} c_i l(\theta, X_i)$ (i.e. weighted empirical loss of each user). In general, we can use any DP ERM algorithms for the new loss l'. For concreteness, we use Algorithm 1 of [BST14] which achieves nearly optimal empirical risk bound.

Algorithm 2 Weighted ERM

Input: $X_1, ..., X_n \in \mathcal{X}$, loss function l,
Parameters: $c_1, \ldots, c_n \in \mathbb{R}_{\geq 0}$, with $\sum_{i=1}^n c_i = 1$
1: Define user-level weighted loss function l' as stated in the above paragraph
2: Run (ε, δ)-DP-ERM algorithm (Algorithm 1 of [BST14]) over user-level loss l' and m users, and obtain its output $\tilde{\theta}$
3: **return** $\tilde{\theta}$

Theorem 4. *Algorithm 2 is (ε, δ)-DP and $\mathbb{E}_{D\sim\mu^n, alg}[L_\mu(\tilde{\theta})] - L_\mu(\theta^*)$ is bounded by*

$$O\left(LB\sqrt{d}\cdot\sqrt{\frac{\log^4(m/\delta)\left(\max_{j=1}^m \sum_{i\in S_j} c_i\right)^2}{\varepsilon^2} + \log(d)\log(n)\sum_{i=1}^n c_i^2}\right).$$

To prove Theorem 4, we prove a weighted version of the uniform convergence result (Theorem 5 of [SSSS09]) to give an upper bound on the generalization error of our weighted ERM algorithm and this result might be of independent interest.

Theorem 5 (Weighted uniform convergence)**.** *For any $\gamma > 0$ and non-negative weights $c = (c_1, ..., c_n)$ with $\sum_{i=1}^n c_i = 1$, with probability at least $1-\gamma$ over $D \sim \mu^n$, we have*

$$\sup_{\theta\in\mathcal{C}} |\hat{L}(\theta, c, D) - L_\mu(\theta)| \leq O\left(LB\sqrt{d\log(d/\gamma)\log(n)\sum_{i=1}^n c_i^2}\right).$$

Tradeoff weights. In ERM and also estimating quantiles (details in the supplementary material), for optimizing the algorithm performance, we need to pick weights to minimize a formula in the form of $\left(\max_{j=1}^m \sum_{i\in S_j} c_i\right)^2 + A\cdot\sum_{i=1}^n c_i^2$, where A depends on parameters of the problem (for example, in Theorem 4, by simply moving terms around, A would be $\Theta(\log(d)\log(n)\varepsilon^2/\log^4(m/\delta))$). This general form intuitively explains the tradeoff we need to make: $A\cdot\sum_{i=1}^n c_i^2$ measures how hard it is to optimize the weighted objective with a user-level private algorithm and $\left(\max_{j=1}^m \sum_{i\in S_j} c_i\right)^2$ measures how well the weighted objective generalizes.

It is not hard to see that Var $\left(\tilde{\beta}\right)$ in Section 3 is also in this form, and the results about characterizing the minimizer and comparisons to the sample limiting algorithm also apply here. And we can also use the same method to compute weights $c = (c_1, ..., c_n)$.

5 Linear regression with label privacy

In this section, we consider linear regression. We are given n data points of the form (X_i, y_i), where $X_i \in \mathbb{R}^d$ and $y_i \in [0, B]$ for $i = 1, \ldots, n$. The y_i values are generated as $y_i = \beta \cdot X_i + \xi_i$, for a vector $\beta \in \mathbb{R}^d$ unknown to us, and random variables ξ_i representing noise. These random variables are assumed to be independent, each with a mean of 0 and a variance of σ^2. We provide a detailed preliminaries to linear regression in the supplementary matirel.

We focus on label-privacy (introduced in [CH11]) in which we protect the privacy of label y_i's and X_i's are public data. In other words, D and D' are considered to be neighboring databases if they have the same X_i's and their y_i's are also the same except data points from just one user l.

Our goal is to have an ε-DP algorithm which estimate β by outputting an unbiased estimator $\tilde{\beta}$ and minimizes the squared error $\mathbb{E}\left[\sum_{j=1}^d (\beta_j - \tilde{\beta}_j)^2\right]$. Since $\tilde{\beta}$ is unbiased, minimizing the squared error is equivalent to minimizing the variance $\sum_{j=1}^d \text{Var}(\tilde{\beta}_j)$.

5.1 The Algorithm

The weighted averaging algorithm of Section 3.1 generalizes the simple averaging algorithm and is a generic linear unbiased estimator for the simple problem. Similarly, for linear regression, we need a generalization of the OLS (ordinary least squared) estimator that provides a generic linear unbiased estimator for β. Such a generalization has already been proposed by [Ait34], albeit for a different purpose[1].

As a fact (see the supplementary material for details), any linear unbiased estimator can be written as $\hat{\beta} = Cy$, for a $d \times n$ matrix $C = [c_{i,j}]_{d\times n}$ satisfying $CX = I_d$. Our generalization of the weighted averaging algorithm to the higher-dimensional case is to use such an estimator followed by the Laplace mechanism:

Algorithm 3 Generalized Weighted Averaging GWA_C

Input: $X \in \mathbb{R}^{n\times d}, y \in \mathbb{R}^{n\times 1}$
Parameters: $C \in \mathbb{R}^{d\times n}$ satisfying $CX = I_d$
1: $\hat{\beta} \leftarrow Cy$
2: $b \leftarrow B\varepsilon^{-1} \cdot \max_l \sum_{j=1}^{d} \sum_{i\in S_l} |c_{j,i}|$
3: independently draw values $W_1, \ldots, W_d$ from Lap(b)
4: $\tilde{\beta} \leftarrow \hat{\beta} + (W_1, \ldots, W_d)$
5: **return** $\tilde{\beta}$

In the following theorem, we provide the performance of our algorithm and its proof can be found in the supplementary material.

Theorem 6. *For every $d \times n$ matrix C satisfying $CX = I_d$, the algorithm GWA_C is ε-DP, and the total variance $\sum_{j=1}^{d} Var(\tilde{\beta}_j)$ of the vector $\tilde{\beta}$ produced by the algorithm GWA_C can be written as:*

$$\sigma^2 \sum_{j=1}^{d}\sum_{i=1}^{n} c_{j,i}^2 + 2d\left(B\varepsilon^{-1} \cdot \max_l \sum_{j=1}^{d}\sum_{i\in S_l} |c_{j,i}|\right)^2.$$

This total variance is a convex function and can be minimized in polynomial time.

5.2 The sample limiting algorithm

We generalize the sample limiting algorithm in Section 3.2 to higher dimensions. The sample limiting algorithm is parameterized by an integer threshold h. For each user l, it randomly picks $\min(h, s_l)$ data points from the user. Let U be the features and v be the labels of the sample data points. The sample limiting algorithm first computes the OLS of the sample data $\hat{\beta}^s = (U^TU)^{-1}U^Tv$. Let $C^s = (U^TU)^{-1}U^T$. For each user $l \in [m]$, let S_l^* be the set of corresponding row numbers in U and v. The algorithm finally outputs a vector $\tilde{\beta}^s$ obtained by adding to each entry of $\hat{\beta}^s$ a value drawn i.i.d. from Lap(b), for $b = B\varepsilon^{-1} \cdot \max_{l=1}^{m} \sum_{j=1}^{d} \sum_{i\in S_l^s} |c_{j,i}^s|$.

Similarly to Section 3.2, after fixing the selected points, the sample limiting algorithm can be considered as a special case of the GWA_C algorithm if we expand C^s to n rows. Therefore the output of the sample limiting algorithm is ε-DP and $\sum_{j=1}^{d} \text{Var}(\tilde{\beta}_j^s)$ is

$$\sigma^2 \sum_{j=1}^{d}\sum_{i=1}^{n} {c_{j,i}^s}^2 + 2d\left(B\varepsilon^{-1} \cdot \max_{l=1}^{m} \sum_{j=1}^{d}\sum_{i\in S_l} |c_{j,i}^s|\right)^2.$$

5.3 An unbounded gap

Here we give an example which shows that the optimal GWA algorithm can have a much smaller variance than the sample limiting algorithm. This advantage of GWA algorithm can also be seen in the experiments mentioned in Section 6.

[1] [Ait34] proposed the Generalized Least Squares method to solve linear regression when the noise in different observations are correlated.

In this example (Example 1), data points are from two orthogonal directions (i.e. X_i's have either $X_{i,1} = 0$ or $X_{i,2} = 0$). To control the user contributions, the sample limiting algorithm wants to pick h big for the first dimension and to pick h small for the second dimension. The sample limiting algorithm has to pick the same threshold h for both dimensions. Intuitively, it cannot avoid big user contributions. A formal proof is provided in the supplementary material.

Example 1. *$d = 2$. Set $\sigma = 0$ and $2d(B/\varepsilon)^2 = 1$. Let g be some integer parameter and set the number of users to be $m = 2g^2 + 2$. Now consider the data points of users:*

- *User 1 has 1 data point with $X_i = (g, 0)$.*
- *Each of user $2, ..., g^2 + 1$ has g data points with same $X_i = (1, 0)$*
- *User $g^2 + 2$ has g data points with same $X_i = (0, 1)$.*
- *Each of user $g^2 + 3, ..., 2g^2 + 2$ has 1 data point with $X_i = (0, 1)$*

Claim 3. *In Example 1, the minimum variance of the sample limiting algorithm is at least $1/4g^3$ and the minimum variance of the generalized weighted averaging algorithm is at most $1/g^4$.*

Example 1 contained one data point whose norm of X_i is much larger than the others. We provide a different example in the supplementary material to show that a gap still exists even when $|X_i| = 1$ for all data points.

6 Experiments

In this section we perform an empirical evaluation of our algorithm and we compare it with the sample limiting algorithm for linear regression in the label-privacy case. In Appendix E, we also provide experimental results on logistic regression using our ERM algorithm of Setion 4.

Datasets We evaluated all methods on two *publicly-available* datasets containing real-world data as well as synthetic datasets with ground-truth generated with standard open-source libraries. We stress that no private data has been used in these experiments. We only briefly describe our and experimental setup here, more details are available in Appendix E.

Synthetic data: We generated regression problem instances with `sklearn`'s `make_regression` ($n \in [600, 3000]$ samples, $d = 10$ features, bias=0.0 and noise=20). To model user contributions we used the Zipf's (power law) distribution for the number of rows of a user (users contributions are often heavy tailed [AH02]). **Real-world datasets:** We used also two UCI Machine Learning Datasets. **drugs** [GKMZ18] ($n = 3107$, $d = 8$, $m = 502$ users with min 1 and max 63 samples) and **news** [MT18] ($n = 3452$, $d = 10$, $m = 297$ users with min 1 and max 878 samples).

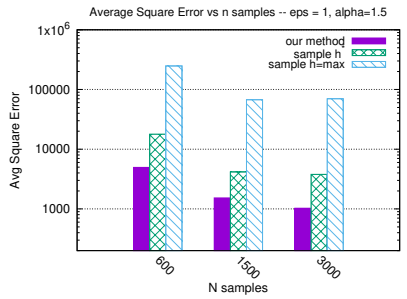


Figure 1: Average squared errors—synthetic dataset, $\alpha = 1.5$ and $\varepsilon = 1$.

Experimental set up Experiments are repeated 10 times and we report mean of each metric computed. For quality we use the average squared error for the prediction. We evaluate our general setting algorithm in Section 5 using $\varepsilon = 1, 2, 3$ values. For σ^2, we treat it as public knowledge and compute it with OLS regression. For **sample limiting** we evaluate the best threshold h^* and the whole datasets (i.e. h set to max user contribution).

Results on the synthetic dataset We compare all methods on datasets with varying numbers of samples n and different parameter α of the Zipf's distribution. Lower α values correspond to more uneven distributions (i.e. some users may many more data points than others). The results for $\alpha = 1.5$, $\varepsilon = 1$, are plot in Figures 1. As expected, the larger the number of samples the lower the loss of all methods, however in every setting our method has always significantly lower

Dataset	ε	Our method	Sample limit h^*	Sample limit h_{max}
drugs	1	3.1	24.8	95.4
	2	2.5	7.7	25.2
	3	2.3	4.5	12.4
news	1	1696.3	96344.4	4862670.7
	2	440.1	24110.0	1201568.9
	3	166.2	10648.3	550989.2

Table 1: Average squared errors for our method, sample limit with best threshold (h^*), and using all data (h_max).

squared error, even orders of magnitude lower (notice the y-axis is in log scale). We now fix $\varepsilon = 1$ and $n = 3000$ samples and analyze the effect of the parameter α in Figure 2. Recall that α controls the inequality in the distribution of the user's contributions. As expected, our method is comparatively much better for low α (i.e., more unequal distributions of user contributions), but it performs always better.

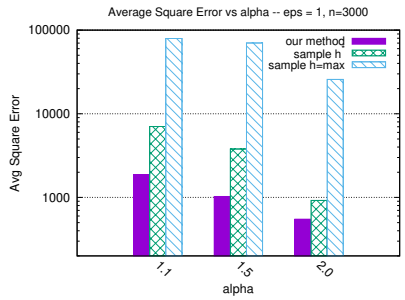


Figure 2: Average squared errors—Synthetic dataset, $\alpha = 1.5$, $\varepsilon = 1, n = 3000$.

Results on real-world datasets We now report the results for the real-world datasets. Our results are summarized in Table 1. The results confirm all empirical observations on the synthetic datasets: the loss decreases for increasing ε for all methods, but our method has always significantly lower loss than both sample limiting with best and max threshold. Notice that the squared error is overall larger for news than for drugs, this is explained by the larger range of the y's values (in news the values are in $[0, 71]$ vs $[1, 10]$ for drugs).

7 Conclusion

In this paper, we propose the weighted averaging method for smoothly bounding user contribution in differential privacy. We apply this method to estimating the mean and quantiles, empirical risk minimization, and linear regression. We show it has advantage over the sampling limiting algorithm, especially in the label-privacy case.

Broader Impact

Privacy is a fundamental concern in machine learning. Respecting the privacy of the users is a requirement of any real system and differential privacy allows to formalize such requirement. In this paper we provided algorithms with improved trade-offs of utility vs differential privacy. This may enable better outcomes for the users of a system at the same level of privacy. We stress that privacy is only one of the requirements of a real system. Any machine learning technology must also responsibly ensure utility of the system and fairness of the system to the users. Privacy requirements may negatively affect utility, and it is known that differential privacy potentially disparately impacts certain users [BPS19]. Such considerations are beyond the scope of the paper and we refer to the emerging literature on responsible machine learning for addressing them [KR19].

Funding Transparency Statement

No third-party funding has been used for this research.

References

[AGK17] Mohammad Alaggan, Sébastien Gambs, and Anne-Marie Kermarrec. Heterogeneous differential privacy. *Journal of Privacy and Confidentiality*, 7(2), Jan. 2017.

[AH02] Lada A Adamic and Bernardo A Huberman. Zipf's law and the internet. *Glottometrics*, 3(1):143–150, 2002.

[Ait34] AC Aitken. On Least-squares and Linear Combinations of Observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1934.

[AKMV19] Kareem Amin, Alex Kulesza, Andres Munoz, and Sergei Vassilvitskii. Bounding user contributions: A bias-variance trade-off in differential privacy. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 263–271, June 2019.

[BFTT19] Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 11279–11288, 2019.

[BPS19] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In *Advances in Neural Information Processing Systems 32*, pages 15479–15488. 2019.

[BST14] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, FOCS '14, page 464–473, USA, 2014. IEEE Computer Society.

[CH11] Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. In *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pages 155–186, June 2011.

[CM08] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, page 289–296, Red Hook, NY, USA, 2008. Curran Associates Inc.

[CMS11] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *J. Mach. Learn. Res.*, 12(null):1069–1109, July 2011.

[DJW13] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438, 2013.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, page 265–284, 2006.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014.

[Dwo06] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[FKT20] Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: Optimal rates in linear time. In *STOC*, 2020.

[Gau26] Carl Friedrich Gauss. *Theoria Combinationis Observationum Erroribus Minimis Obnoxiae, Parts 1, 2 and suppl. Werke 4, 1–108*. 1821, 1823, 1826.

[GKMZ18] Felix Gräßer, Surya Kallumadi, Hagen Malberg, and Sebastian Zaunseder. Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In *Proceedings of the 2018 International Conference on Digital Health*, pages 121–125, 2018.

[INS+19] R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 299–316, 2019.

[JKT12] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of*

Machine Learning Research, pages 24.1–24.34, Edinburgh, Scotland, 25–27 Jun 2012. PMLR.

[JT14] Prateek Jain and Abhradeep Guha Thakurta. (near) dimension independent risk bounds for differentially private learning. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 476–484, Bejing, China, 22–24 Jun 2014. PMLR.

[JYC15] Z. Jorgensen, T. Yu, and G. Cormode. Conservative or liberal? personalized differential privacy. In *2015 IEEE 31st International Conference on Data Engineering*, pages 1023–1034, 2015.

[KMA+19] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *CoRR*, abs/1912.04977, 2019.

[KR19] Michael Kearns and Aaron Roth. *The Ethical Algorithm: The Science of Socially Aware Algorithm Design*. Oxford University Press, 2019.

[KST12] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pages 25.1–25.40, Edinburgh, Scotland, 25–27 Jun 2012. PMLR.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, page 94–103, USA, 2007. IEEE Computer Society.

[MT18] Nuno Moniz and Luis Torgo. Multi-source social feedback of online news feeds. *CoRR*, 2018.

[PSY+19] Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X. Yu, Sashank J. Reddi, and Sanjiv Kumar. Adaclip: Adaptive clipping for private SGD. *CoRR*, abs/1908.07643, 2019.

[SCS13] S. Song, K. Chaudhuri, and A. D. Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248, 2013.

[SCS15] Shuang Song, Kamalika Chaudhuri, and Anand Sarwate. Learning from Data with Heterogeneous Noise using SGD. volume 38 of *Proceedings of Machine Learning Research*, pages 894–902, San Diego, California, USA, 09–12 May 2015. PMLR.

[She19a] Or Sheffet. Differentially private ordinary least squares. *Journal of Privacy and Confidentiality*, 9(1), Mar. 2019.

[She19b] Or Sheffet. Old techniques in differentially private linear regression. In Aurélien Garivier and Satyen Kale, editors, *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98 of *Proceedings of Machine Learning Research*, pages 789–827, Chicago, Illinois, 22–24 Mar 2019. PMLR.

[Smi11] Adam Smith. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 813–822, New York, NY, USA, 2011. Association for Computing Machinery.

[SSSS09] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009.

[STU17] A. Smith, A. Thakurta, and J. Upadhyay. Is interaction necessary for distributed private learning? In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 58–77, 2017.

[TAM19] Om Thakkar, Galen Andrew, and H. Brendan McMahan. Differentially private learning with adaptive clipping. *CoRR*, abs/1905.03871, 2019.

[TS13] Abhradeep Guha Thakurta and Adam Smith. Differentially private feature selection via stability arguments, and the robustness of the lasso. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 819–850, Princeton, NJ, USA, 12–14 Jun 2013. PMLR.

[TTZ15] Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Nearly-optimal private lasso. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 3025–3033, Cambridge, MA, USA, 2015. MIT Press.

[Ull15] Jonathan Ullman. Private multiplicative weights beyond linear queries. In *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS '15, page 303–312, New York, NY, USA, 2015. Association for Computing Machinery.

[Wan18] Yu-Xiang Wang. Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence*, pages 93–103, 2018.

[WLK+17] Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, page 1307–1322, New York, NY, USA, 2017. Association for Computing Machinery.

[WX19] Di Wang and Jinhui Xu. On sparse linear regression in the local differential privacy model. volume 97 of *Proceedings of Machine Learning Research*, pages 6628–6637, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[WYX17] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2722–2731. Curran Associates, Inc., 2017.

[ZZMW17] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private ERM for smooth objectives. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 3922–3928. ijcai.org, 2017.

A Missing proofs of Section 3

Lemma 4 (Restatement of Lemma 3)**.** *Let $c^* = (c_1^*, ..., c_n^*)$ be the vector that minimizes Var($\tilde{\beta}$). There exists h, such that,*

1. $s_1 \leq h \leq s_m$
2. *For each data point i of user q, $c_i^* = \frac{\min(h, s_q)}{s_q \sum_{l=1}^m \min(h, s_l)}$.*

Proof. We show in this proof how to pick the h as stated in the lemma. Let M be the set $\arg\max_{l=1}^m \sum_{i \in S_l} c_i^*$. We start by considering any two data points: data point i of user p and data point i' from of q. Consider the following cases:

- $q \in M$ and $p \notin M$: In this case, $c_i^* \leq c_{i'}^*$. If not, pick $\eta = \min((c_{i'}^* - c_i^*)/2, (\max_l \sum_{i \in S_l} c_i^*) - \sum_{i \in S_p} c_i^*)$. Changing c_i^* and $c_{i'}^*$ to $c_i^* + \eta$ and $c_{i'}^* - \eta$ would decrease $\sigma^2((c_1^*)^2 + \cdots + (c_n^*)^2)$ and would not change $\max_l \sum_{i \in S_l} c_i^*$. Therefore this change would decrease Var($\tilde{\beta}$) and result in a contradiction.
- $p \neq q$ and $p, q \notin M$: In this case $c_i^* = c_{i'}^*$. If not, without loss of generality, assume $c_i^* < c_{i'}^*$. Pick $\eta = \min((c_{i'}^* - c_i^*)/2, (\max_l \sum_{i \in S_l} c_i^*) - \sum_{i \in S_p} c_i^*)$. Similarly as the previous case, changing c_i^* and $c_{i'}^*$ to $c_i^* + \eta$ and $c_{i'}^* - \eta$ would decrease Var($\tilde{\beta}$) and result in a contradiction.
- $p = q$: In this case $c_i^* = c_{i'}^*$. If not, changing both c_i^* and $c_{i'}^*$ to $(c_i^* + c_{i'}^*)/2$ would decrease Var($\tilde{\beta}$) and result in a contradiction.

With the above characterization, we are ready to pick h. We start with a special case in which $M = [m]$. In this case, $\sum_{i \in S_l} c_i^*$ are the same for all users. Therefore $\sum_{i \in S_l} c_i^* = 1/m$ for any l. We also know from the above that data points of the same user have the same c_i^*. Therefore, data point i of user q has $c_i^* = \frac{1}{m \cdot s_l}$. Setting $h = s_1$ would work, since

$$\frac{\min(h, s_q)}{s_q \sum_{l=1}^m \min(h, s_l)} = \frac{h}{s_l \cdot m \cdot h} = \frac{1}{m \cdot s_l} = c_i^*.$$

Now consider the case in which $M \neq [m]$. Pick an arbitrary user $p \notin M$ and some $i_p \in S_p$. Set $h = (\max_{l=1}^m \sum_{i \in S_l} c_i^*)/c_{i_p}^*$. We are going to show such h works as the lemma statement.

For any user $q \in M$ and for any $i \in S_q$, using the above characterization, we know $c_i^* \leq c_{i_p}^*$. Therefore

$$h = \frac{\max_{l=1}^m \sum_{i \in S_l} c_i^*}{c_{i_p}^*} = \frac{\sum_{i \in S_q} c_i^*}{c_{i_p}^*} \leq s_q.$$

For any user $q \notin M$ and for any $i \in S_q$, using the above characterization, we know $c_i^* = c_{i_p}^*$. Therefore

$$h = \frac{\max_{l=1}^m \sum_{i \in S_l} c_i^*}{c_{i_p}^*} > \frac{\sum_{i \in S_q} c_i^*}{c_{i_p}^*} = s_q.$$

So we have $\sum_{l=1}^m \min(h, s_l) = |M|h + \sum_{l \notin M} s_l$. Using the above characterization of two data points, we have

$$\left(|M|h + \sum_{l \notin M} s_l\right) \cdot c_{i_p}^* = \left(\sum_{l \in M} \sum_{i \in S_l} c_i^*\right) + \left(\sum_{l \notin M} \sum_{i \in S_l} c_i^*\right)$$
$$= 1.$$

For any user $q \in M$ and for any $i \in S_q$,

$$\frac{\min(h, s_q)}{s_q \sum_{l=1}^{m} \min(h, s_l)} = \frac{h}{s_q(|M|h + \sum_{l \notin M} s_l)}$$
$$= \frac{h \cdot c^*_{i_p}}{s_q} = \frac{\sum_{i' \in S_q} c^*_i}{s_q} = c^*_i.$$

For any user $q \notin M$ and for any $i \in S_q$,

$$\frac{\min(h, s_q)}{s_q \sum_{l=1}^{m} \min(h, s_l)} = \frac{1}{\sum_{l=1}^{m} \min(h, s_l)} = c^*_{i_p} = c^*_i.$$

□

Theorem 7 (Restatement of Theorem 2)**.** *For every $g \in \mathbb{N}$, there is an instance where the variance of the best weighted averaging algorithm is less than $\frac{2g+1}{4g^2}$, while the variance of the best sample limiting algorithm is at least $\frac{3}{4g}$. For every g, $\frac{3}{4g} \geq \frac{2g+1}{4g^2}$, and the ratio between these two numbers converges to $3/2$ as g goes to infinity.*

Proof. We use the following example:

Example 2. *For some $g \in \mathbb{N}$, set $m = 2g$. Set $s_l = 1$ for $l = 1, ..., g$ and $s_l = g$ for $l = g+1, ..., 2g$. Set $\sigma = 1$ and $\frac{B}{\varepsilon} = \sqrt{g/2}$.*

The variance of the weighted averaging algorithm can be bounded by:

$$\min_{h: s_1 \leq h \leq s_m} v(h) \leq v(1) = \frac{2g+1}{4g^2}.$$

For sample limiting, for every $h \geq 1$, we have

$$v'(h) = \frac{1}{g + g \cdot h} + g \left(\frac{h}{g + g \cdot h} \right)^2 = g \left(\frac{h}{g + g \cdot h} - \frac{1}{2g} \right)^2 - \frac{1}{4g} + \frac{1}{g} \geq \frac{3}{4g}.$$

Therefore, the variance of the best sample limiting algorithm is $\min_{h: s_1 \leq h \leq s_m, h \in \mathbb{N}} v'(h) \geq \frac{3}{4g}$. □

Theorem 8 (Restatement of Theorem 3)**.** *In every instance, the variance of the best sample limiting algorithm is at most 4 times the variance of the best weighted averaging algorithm.*

Proof. We need to prove $\min_{h: s_1 \leq h \leq s_m, h \in \mathbb{N}} v'(h) \leq 4 \min_{h: s_1 \leq h \leq s_m} v(h)$. Let $h^* = \arg\min_{h: s_l \leq h \leq s_m} v(h)$. Let p be some index such that $s_p \leq h^* \leq s_{p+1}$. Let $q = m - p$. There are at least q users with $s_l \geq h^*$. Set $A = \sum_{l=1}^{p} s_l$. Set $\alpha = A/n_{h^*}$. We consider two cases:

Case 1 ($\alpha \geq 1/2$): In this case we set $h' = \lceil h^* \rceil$ and we want to show $v'(h')$ is not much bigger than $v(h^*)$.

First of all, since $h^* \leq h'$, we know that $n_{h^*} \leq n_{h'}$. We have

$$\sigma^2 \sum_l s_l \cdot \left(\frac{\min(h^*, s_l)}{n_{h^*} \cdot s_l} \right)^2 \geq \sigma^2 \sum_{l: h^* \geq s_l} s_l \cdot \left(\frac{\min(h^*, s_l)}{n_{h^*} \cdot s_l} \right)^2 = \sigma^2 \cdot A \cdot \left(\frac{1}{n_{h^*}} \right)^2 \geq \frac{\sigma^2}{2n_{h^*}} \geq \frac{\sigma^2}{2n_{h'}}.$$

On the other hand, we know $h^* \geq s_1 \geq 1$, and therefore $2h^* \geq \lceil h^* \rceil = h'$. So we have $\left(\frac{B \cdot h'}{\varepsilon \cdot n_{h'}} \right)^2 \leq \left(\frac{B \cdot 2h^*}{\varepsilon \cdot n_{h^*}} \right)^2 = 4 \cdot \left(\frac{B \cdot h^*}{\varepsilon \cdot n_{h^*}} \right)^2$. Putting things together, we get

$$v'(h') = \frac{\sigma^2}{n_{h'}} + 2 \left(\frac{B \cdot h'}{\varepsilon \cdot n_{h'}} \right)^2 \leq 2\sigma^2 \sum_l s_l \cdot \left(\frac{\min(h^*, s_l)}{n_{h^*} \cdot s_l} \right)^2 + 4 \cdot 2 \left(\frac{B \cdot h^*}{\varepsilon \cdot n_{h^*}} \right)^2 \leq 4v(h^*).$$

Case 2 ($\alpha < 1/2$): In this case, setting h' close to h might not work. We pick $\eta = 1 - \frac{1}{2(1-\alpha)}$ and $r = \lceil q \cdot \eta \rceil$, and set $h' = s_{p+r}$.

We want to show that $v'(h') \le 4v(h^*)$. Let's start with the first parts of $v'(h')$ and $v(h^*)$. We have

$$\sigma^2 \sum_{l=1}^{m} s_l \cdot \left(\frac{\min(h^*, s_l)}{n_{h^*} \cdot s_l}\right)^2 \ge \sigma^2 \sum_{l=1}^{p+r} s_l \cdot \left(\frac{\min(h^*, s_l)}{n_{h^*} \cdot s_l}\right)^2 \ge \sigma^2 \cdot \frac{\left(\frac{\sum_{l=1}^{p+r} \min(h^*, s_l)}{n_{h^*}}\right)^2}{\sum_{l=1}^{p+r} s_l}$$
$$= \sigma^2 \cdot \frac{(\alpha + (1-\alpha) * \eta)^2}{\sum_{l=1}^{p+r} s_l} = \frac{\sigma^2}{4\sum_{l=1}^{p+r} s_l} = \frac{v'(h')}{4}.$$

Now we consider the second parts of $v'(h')$ and $v(h^*)$. We have $\frac{h^*}{n_{h^*}} = \frac{n_{h^*} - A}{q \cdot n_{h^*}} = \frac{1-\alpha}{q}$ and

$$\frac{h'}{n_{h'}} \le \frac{h'}{\sum_{l=p+r}^{p+q} \min(h', s_l)} = \frac{h'}{(q-r+1)h'} \le 1/(1-\eta).$$

Therefore $2\left(\frac{B \cdot h'}{\varepsilon \cdot n_{h'}}\right)^2 \le 4 \cdot 2\left(\frac{B \cdot h^*}{\varepsilon \cdot n_{h^*}}\right)^2$.

To sum up, we get $v'(h') \le 4v(h^*)$. □

B Estimating quantiles

In this section, we show that the idea of our weighted average algorithm can be also applied to estimating quantiles. We extend the PrivateQuantile algorithm of [Smi11] which is mainly based on the exponenetial mechanism [MT07] to the case when a user controls more than one data point.

Here we describe the setting of estimating quantiles. We are given n samples $D = (y_1, ..., y_n)$ sampled independently from an unknown distribution μ supported on real numbers. The goal is to output the q-th quantile of distribution μ.

As prior knowledge, we are given a base distribution ν satisfying the following assumption for some parameters α and β. Here $\mathbb{F}_\mu$ is the cumulative distribution function of μ. This assumption can be interpreted as that elements within α distance in the quantile space to the q-th quantile of μ have probability density at least β in the base distribution ν.

Assumption 1. *Define set $S_{q,\alpha} = \left\{y \middle| \ q - \alpha \le \mathbb{F}_\mu(y) \le q + \alpha\right\}$. We have $\nu(S_{q,\alpha}) \ge \beta$.*

Now we describe our algorithm for estimating quantiles. The main idea is to switch the rank function used in PrivateQuantile of [Smi11] to a weighted rank function parametrized by weights $c = (c_1, ..., c_n)$ and apply the exponential mechanism[MT07].

Algorithm 4 Weighted Rank WR_c

Input: $y_1, \ldots, y_n \in \mathbb{R}$, $q \in (0, 1)$, base distribution ν
Parameters: $c_1, \ldots, c_n \in \mathbb{R}_{\ge 0}$, with $\sum_{i=1}^n c_i = 1$
1: Define the weighted rank function **weighted-rank**$(y, D) = \sum_{i=1}^n c_i \cdot \mathbb{1}\{y \ge y_i\}$.
2: Set W to be the maximum total weights of a single user, i.e. $W = \max_{j=1}^m \sum_{i \in S_j} c_i$.
3: Sample $\tilde{y}$ with probability proportional to $\nu(y) \cdot \exp\left(-\frac{\varepsilon}{2W} \cdot |\textbf{weighted-rank}(y, D) - q|\right)$
4: **return** $\tilde{y}$

Theorem 9. *Algorithm 4 is ε-DP and with probability at least $1 - 2\gamma$,*

$$|\mathbb{F}_\mu(\tilde{y}) - q| = O\left(\alpha + \sqrt{\frac{(\max_{j=1}^m \sum_{i \in S_j} c_i)^2}{\varepsilon^2} \ln^2\left(\frac{1}{\gamma\beta}\right) + \ln\left(\frac{n}{\gamma}\right) \cdot \sum_{i=1}^n c_i^2}\right).$$

In this section, we prove Theorem 9. Its privacy guarantee is proved in Claim 4 and its utility guarantee is proved in Corollary 1.

We first prove the privacy guarantee of Algorithm 4.

Claim 4. *Algorithm 4 is ε-DP.*

Proof. For any neighboring dataset D and D' and any $y \in \mathbb{R}$, we know the weighted rank function can be changed by at most the maximum total weight of a single user, i.e.

$$|\textbf{weighted-rank}(y, D) - \textbf{weighted-rank}(y, D')| \le W = \max_{j=1}^{m} \sum_{i \in S_j} c_i.$$

Therefore,

$$\left| \left(-\frac{1}{2W} \cdot |\textbf{weighted-rank}(y, D) - q| \right) - \left(-\frac{1}{2W} \cdot |\textbf{weighted-rank}(y, D') - q| \right) \right| \le \frac{1}{2}.$$

Apply Theorem 6 of [MT07], we know WR_c is ε-DP. □

We prove two claims before we proceed to Corollary 1.

Claim 5. *With probability $1 - \gamma$, the sampled dataset D has*

$$\sup_{y \in \mathbb{R}} |F_\mu(y) - \textbf{\textit{weighted-rank}}(y, D)| \le \eta = \sqrt{\frac{1}{2} \ln\left(\frac{2n}{\gamma}\right) \cdot \sum_{i=1}^{n} c_i^2} + \max_{i \in [n]} c_i.$$

Proof. We start by bounding

$$\Pr_{D \sim \mu^n} \left[\max_{i \in [n]} |F_\mu(y_i) - \textbf{weighted-rank}(y_i, D)| \le \sqrt{\frac{1}{2} \ln\left(\frac{2n}{\gamma}\right) \cdot \sum_{i=1}^{n} c_i^2} \right].$$

By Hoeffding's inequality, we have for each $i \in [n]$,

$$\Pr_{D \sim \mu^n} \left[|F_\mu(y_i) - \textbf{weighted-rank}(y_i, D)| \le \sqrt{\frac{1}{2} \ln\left(\frac{2n}{\gamma}\right) \cdot \sum_{i=1}^{n} c_i^2} \right] \ge 1 - \frac{\gamma}{n}.$$

By union bound, we get

$$\Pr_{D \sim \mu^n} \left[\max_{i \in [n]} |F_\mu(y_i) - \textbf{weighted-rank}(y_i, D)| \le \sqrt{\frac{1}{2} \ln\left(\frac{2n}{\gamma}\right) \cdot \sum_{i=1}^{n} c_i^2} \right] \ge 1 - \gamma.$$

In the rest of the proof, it suffices to show

$$\sup_{y \in \mathbb{R}} |F_\mu(y) - \textbf{weighted-rank}(y, D)| \le \max_{i \in [n]} |F_\mu(y_i) - \textbf{weighted-rank}(y_i, D)| + \max_{i \in [n]} c_i.$$

There are three cases:

- $y < y_1$. In this case,

$$\begin{aligned} & |F_\mu(y) - \textbf{weighted-rank}(y, D)| \\ =& F_\mu(y) \le F_\mu(y_1) \\ \le& |F_\mu(y_1) - \textbf{weighted-rank}(y_1, D)| + \textbf{weighted-rank}(y_1, D) \\ =& |F_\mu(y_1) - \textbf{weighted-rank}(y_1, D)| + c_1 \\ \le& \max_{i \in [n]} |F_\mu(y_i) - \textbf{weighted-rank}(y_i, D)| + \max_{i \in [n]} c_i. \end{aligned}$$

- $y_i \le y < y_{i+1}$ for some $i \in [n-1]$. Then we have

$$\begin{aligned} & \textbf{weighted-rank}(y, D) - F_\mu(y) \\ =& \textbf{weighted-rank}(y_i, D) - F_\mu(y) \\ \le& \textbf{weighted-rank}(y_i, D) - F_\mu(y_i) \\ \le& |F_\mu(y_i) - \textbf{weighted-rank}(y_i, D)|. \end{aligned}$$

and

$$\begin{aligned}
&F_\mu(y) - \textbf{weighted-rank}(y, D) \\
\leq &F_\mu(y_{i+1}) - \textbf{weighted-rank}(y_i, D) \\
= &F_\mu(y_{i+1}) - \textbf{weighted-rank}(y_{i+1}, D) + c_{i+1} \\
\leq &|F_\mu(y_{i+1}) - \textbf{weighted-rank}(y_{i+1}, D)| + c_{i+1}.
\end{aligned}$$

In this case we also have $|F_\mu(y) - \textbf{weighted-rank}(y, D)| \leq \max_{i\in[n]} |F_\mu(y_i) - \textbf{weighted-rank}(y_i, D)| + \max_{i\in[n]} c_i$..

- $y \geq y_n$. In this case, similarly as the first case, we can show
$$|F_\mu(y) - \textbf{weighted-rank}(y, D)| \leq |F_\mu(y_n) - \textbf{weighted-rank}(y_n, D)| + c_n.$$

□

Claim 6. *Given the sampled dataset D and the fact that the event in Claim 5 is satisfied, with probability at least $1-\gamma$, the output $\tilde{y}$ has*

$$|\textbf{\textit{weighted-rank}}(\tilde{y}, D) - q| \leq \alpha + \eta + \frac{2W}{\varepsilon} \ln\left(\frac{1}{\gamma\beta}\right).$$

Proof. Define two sets $\textbf{GOOD} = \{y \big| |\textbf{weighted-rank}(y, D) - q| \leq \alpha + \eta\}$ and $\textbf{BAD} = \{y \big| |\textbf{weighted-rank}(y, D) - q| \geq \alpha + \eta + \frac{2W}{\varepsilon} \ln\left(\frac{1}{\gamma\beta}\right)|\}$. We know $S_{q,\alpha} \subseteq \textbf{GOOD}$. Therefore, $\nu(\textbf{GOOD}) \geq \nu(S_{q,\alpha}) \geq \beta$. Then we have

$$\begin{aligned}
\Pr[\tilde{y} \in \textbf{BAD}] &\leq \frac{\Pr[\tilde{y} \in \textbf{BAD}]}{\Pr[\tilde{y} \in \textbf{GOOD}]} \\
&\leq \exp\left(-\frac{\varepsilon}{2W} \cdot \frac{2W}{\varepsilon} \ln\left(\frac{1}{\gamma\beta}\right)\right) \frac{\nu(\textbf{BAD})}{\nu(\textbf{GOOD})} \\
&\leq \gamma\beta \cdot \frac{1}{\beta} = \gamma.
\end{aligned}$$

□

By the above two claims, we get the following corollary which bounds the distance between $\tilde{y}$ and the q-th quantile of μ in the quantile space.

Corollary 1. *With probability at least $1-2\gamma$,*

$$\begin{aligned}
|\mathbb{F}_\mu(\tilde{y}) - q| &= O\left(\alpha + \frac{W}{\varepsilon} \ln\left(\frac{1}{\gamma\beta}\right) + \sqrt{\ln\left(\frac{n}{\gamma}\right) \cdot \sum_{i=1}^n c_i^2}\right) \\
&= O\left(\alpha + \sqrt{\frac{W^2}{\varepsilon^2} \ln^2\left(\frac{1}{\gamma\beta}\right) + \ln\left(\frac{n}{\gamma}\right) \cdot \sum_{i=1}^n c_i^2}\right).
\end{aligned}$$

C Missing proofs of Section 4

We prove Theorem 4 in this section.

We show in the following corollary about the privacy guarantee and the weighted empirical risk of Algorithm 2. For notation convenience, define $\hat{\theta}(c, D)$ to be the minimizer of the weighted empirical risk for dataset D and weights c, i.e. $\hat{\theta}(D) \in \arg\min_{\theta\in\mathcal{C}} \hat{L}(\theta, c, D)$.

Corollary 2. *Algorithm 2 is (ε, δ)-DP and for weighted empirical risk, for any dataset $D = (X_1, ..., X_n)$, we have*

$$\mathbb{E}_{alg}|\hat{L}(\tilde{\theta}, c, D) - \hat{L}(\hat{\theta}(c, D), c, D)| = O\left(\frac{LB\sqrt{d}\log^2(m/\delta)\max_{j=1}^m \sum_{i\in S_j} c_i}{\varepsilon}\right).$$

$\mathbb{E}_{alg}$ means that the expectation is over the randomness of the algorithm.

Proof. The differential privacy guarantee of Algorithm 2 simply follows the differential privacy guarantee of Algorithm 1 in [BST14].

Notice that for each user j, function $l'(\cdot, M_j)$ is $(L \cdot \sum_{i \in S_j} c_i)$-Lipschitz. Applying Theorem 2.4 of [BST14] gives the bound in the corollary. □

We prove the weighted version of the uniform convergence result (Theorem 5 of [SSSS09]) to give an upper bound on the generalization error of our weighted ERM algorithm.

Theorem 10 (Restatement of Theorem 5)**.** *For any $\gamma > 0$, with probability at least $1 - \gamma$ over $D \sim \mu^n$, we have*

$$\sup_{\theta \in \mathcal{C}} |\hat{L}(\theta, c, D) - L_\mu(\theta)| \leq O\left(LB\sqrt{d \log(d/\gamma) \log(n) \sum_{i=1}^{n} c_i^2}\right).$$

Proof. By line (10) in Theorem 5 of [SSSS09], we can bound the ℓ_∞-covering of the class of functions $\mathcal{F} = \{X \to l(\theta, X) | \theta \in \mathcal{C}\}$:

$$\mathcal{N}(\alpha, \mathcal{F}, d_\infty) = O\left(d^2 \left(\frac{LB}{\alpha}\right)^d\right).$$

Therefore, there exists a discrete set of $\mathcal{C}' \subset \mathcal{C}$ with size $|\mathcal{C}'| = \mathcal{N}(\alpha, \mathcal{F}, d_\infty) = O\left(d^2 \left(\frac{LB}{\alpha}\right)^d\right)$, such that for any $\theta \in \mathcal{C}$, there exists a $\theta' \in \mathcal{C}'$ satisfying

$$\sup_{X \in \mathcal{X}} |l(\theta, X) - l(\theta', X)| \leq \alpha.$$

Notice that for any $\theta, \theta' \in \mathcal{C}$,

$$\begin{aligned}
|\hat{L}(\theta, c, D) - L_\mu(\theta)| - |\hat{L}(\theta', c, D) - L_\mu(\theta')| &\leq |\hat{L}(\theta, c, D) - \hat{L}(\theta', c, D)| + |L_\mu(\theta) - L_\mu(\theta')| \\
&\leq \left(1 + \sum_{i=1}^{n} c_i\right) \sup_{X \in \mathcal{X}} |l(\theta, X) - l(\theta', X)| \\
&= 2 \sup_{X \in \mathcal{X}} |l(\theta, X) - l(\theta', X)|.
\end{aligned}$$

Therefore, we can focus on the uniform convergence in set $\mathcal{C}'$:

$$\Pr\left[\sup_{\theta \in \mathcal{C}} |\hat{L}(\theta, c, D) - L_\mu(\theta)| \geq 3\alpha\right] \leq \Pr\left[\sup_{\theta \in \mathcal{C}'} |\hat{L}(\theta, c, D) - L_\mu(\theta)| \geq \alpha\right].$$

Recall $\hat{L}(\theta, c, D) = \sum_{i=1}^{n} c_i l(\theta, X_i)$ and $\mathbb{E}_{X_i \sim \mu}[l(\theta, X_i)] = L_\mu(\theta)]$. For each $\theta \in C'$, we can apply Hoeffding's inequality to show that

$$\Pr\left[|\hat{L}(\theta, c, D) - L_\mu(\theta)| \geq \alpha\right] \leq 2 \exp\left(-\frac{2\alpha^2}{\sum_{i=1}^{n} c_i^2 L^2 B^2}\right).$$

Therefore, by union bound over the set $\mathcal{C}'$, we have

$$\begin{aligned}
\Pr\left[\sup_{\theta \in \mathcal{C}'} |\hat{L}(\theta, c, D) - L_\mu(\theta)| \geq \alpha\right] &\leq |\mathcal{C}'| \cdot 2 \exp\left(-\frac{2\alpha^2}{\sum_{i=1}^{n} c_i^2 L^2 B^2}\right) \\
&= O\left(d^2 \left(\frac{LB}{\alpha}\right)^d \exp\left(-\frac{2\alpha^2}{\sum_{i=1}^{n} c_i^2 L^2 B^2}\right)\right).
\end{aligned}$$

Equating the right-hand side to γ gives the bound in the theorem. □

Finally we give the population loss of Algorithm 2.

Corollary 3. *If we run Algorithm 2 with weights $c = (c_1, ..., c_n)$, we can bound $\mathbb{E}_{D\sim\mu^n, alg}[L_\mu(\tilde{\theta})] - L_\mu(\theta^*)$ by*

$$O\left(LB\sqrt{d}\cdot\sqrt{\frac{\log^4(m/\delta)\left(\max_{j=1}^m\sum_{i\in S_j}c_i\right)^2}{\varepsilon^2}+\log(d)\log(n)\sum_{i=1}^n c_i^2}\right).$$

Proof. First notice that we can rewrite $L_\mu(\theta^*)$ as

$$L_\mu(\theta^*) = \mathbb{E}_{X\sim\mu}[l(\theta^*, X)] = \mathbb{E}_{D\sim\mu^n}\left[\sum_{i=1}^n c_i l(\theta^*, X_i)\right] = \mathbb{E}_{D\sim\mu^n}[\hat{L}(\theta^*, c, D)].$$

We prove this corollary by breaking the difference into three parts:

$$\begin{aligned}
&\mathbb{E}_{D\sim\mu^n, alg}[L_\mu(\tilde{\theta})] - L_\mu(\theta^*)\\
=&\mathbb{E}_{D\sim\mu^n, alg}\Bigg[\left(\hat{L}(\hat{\theta}(c,D),c,D) - \hat{L}(\theta^*,c,D)\right) + \left(\hat{L}(\tilde{\theta},c,D) - \hat{L}(\hat{\theta}(c,D),c,D)\right)\\
&\qquad + \left(L_\mu(\tilde{\theta}) - \hat{L}(\tilde{\theta},c,D)\right)\Bigg].
\end{aligned}$$

First by the definition of $\hat{\theta}(c, D)$, we have, for any D,

$$\hat{L}(\hat{\theta}(c,D),c,D) - \hat{L}(\theta^*,c,D) \le 0.$$

For the weighted empirical risk, by Corollary 2, we have

$$\mathbb{E}_{D\sim\mu^n, alg}[\hat{L}(\tilde{\theta},c,D) - \hat{L}(\hat{\theta}(c,D),c,D)] = O\left(\frac{LB\sqrt{d}\log^2(m/\delta)\max_{j=1}^m\sum_{i\in S_j}c_i}{\varepsilon}\right).$$

For the generalization error, by Theorem 5, by picking $\gamma = 1/\sqrt{d}$, we have

$$\mathbb{E}_{D\sim\mu^n, alg}[L_\mu(\tilde{\theta}) - \hat{L}(\tilde{\theta},c,D)] = O\left(LB\sqrt{d\log(d)\log(n)\sum_{i=1}^n c_i^2}\right)$$

To sum up, we get

$$\begin{aligned}
&\mathbb{E}_{D\sim\mu^n, alg}[L_\mu(\tilde{\theta})] - L_\mu(\theta^*)\\
\le& O\left(LB\sqrt{d}\left(\frac{\log^2(m/\delta)\max_{j=1}^m\sum_{i\in S_j}c_i}{\varepsilon} + \sqrt{\log(d)\log(n)\sum_{i=1}^n c_i^2}\right)\right)\\
=& O\left(LB\sqrt{d}\cdot\sqrt{\frac{\log^4(m/\delta)\left(\max_{j=1}^m\sum_{i\in S_j}c_i\right)^2}{\varepsilon^2}+\log(d)\log(n)\sum_{i=1}^n c_i^2}\right).
\end{aligned}$$

□

.

D Missing details and proofs of Section 5

D.1 Linear regression preliminaries

In the linear regression problem, we are given n data points of the form (X_i, y_i), where $X_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ for $i = 1, \ldots, n$. The y_i values are generated as

$$y_i = \beta \cdot X_i + \xi_i.$$

for a vector $\beta \in \mathbb{R}^d$ unknown to us, and random variables ξ_i representing noise. These random variables are assumed to be independent, each with a mean of 0 and a variance of σ^2. Our goal is to estimate the vector β.

A common way to solve a linear regression problem is to find β that minimizes the *empirical loss* $\sum_{i=1}^{n}(\beta \cdot X_i - y_i)^2$. Let X denote the $n \times d$ matrix containing X_i's as its rows, and y denote the $n \times 1$ vector containing y_i's. We assume X has rank d. Then, by taking derivatives and simple linear algebra, minimizing empirical risk corresponds to the following estimator:

Definition 4 (Ordinary Least Squares Estimator (OLS))**.** *The ordinary least squares estimator (OLS) is*

$$\hat{\beta} = (X^T X)^{-1} X^T y.$$

The classical Gauss-Markov theorem shows that OLS is the best among all estimators satisfying two natural properties defined below.

Definition 5 (Linear Estimators)**.** *A linear estimator is an estimator of the form*

$$\hat{\beta} = C \cdot y,$$

where C is a $d \times n$ matrix that can depend on X, but not on y.

Definition 6 (Unbiased Estimators)**.** *An estimator $\hat{\beta}$ is unbiased if for all β and X, the estimates $\hat{\beta}_j$ satisfy*

$$\forall j: \ \mathbb{E}\left[\hat{\beta}_j\right] = \beta_j.$$

Theorem 11 (Gauss-Markov [Gau26])**.** *Let $\hat{\beta}$ be an estimate for β. For a vector $\lambda \in \mathbb{R}^d$, the mean squared error of this estimate at λ is defined as $\mathbb{E}\left[(\beta \cdot \lambda - \hat{\beta} \cdot \lambda)^2\right]$.[2] Then for* every *$\lambda \in \mathbb{R}^d$, among all linear unbiased estimators, OLS has the lowest mean squared error at λ.*

A simple corollary of the above theorem is that among all linear unbiased estimators, OLS is the one with the minimum $\mathbb{E}\left[\sum_{j=1}^{d}(\beta_j - \hat{\beta}_j)^2\right]$.

We will need the following fact, which is part of the proof of the Gauss-Markov theorem.

Fact 1. *Let $\hat{\beta} = Cy$ be a linear estimator. $\hat{\beta}$ is an unbiased estimator if and only if $CX = I_d$, where I_d is the $d \times d$ identity matrix.*

D.2 Linear and unbiased estimators

We give some characterization of linear and unbiased estimators.

When the error random variables ξ_i's (defined in Section D.1) are correlated and have 0 mean and covariance matrix Ω, the following generalized least squares estimator has been shown to be the the BLUE. And we connect generalized least squares estimator to an arbitrary linear and unbiased estimator in Claim 7.

Definition 7 (Generalized Least Squares Estimator (GLS) [Ait34])**.** *The generalized least squares estimator (GLS) is*

$$\hat{\beta} = (X^T \Omega^{-1} X)^{-1} X^T \Omega^{-1} y.$$

Claim 7. *Any linear unbiased estimator is the GLS for some covariance matrix.*

Proof. Let the linear unbiased estimator be Cy. By Fact 1, we know that $CX = I_d$. So $\text{rank}(C) \geq \text{rank}(I_d) = d$ and then $\text{rank}(C^T C) = \text{rank}(C) \geq d$. Therefore $C^T C$ is full-rank and invertible.

Now set $\Omega = (C^T C)^{-1}$, we have

$$\begin{aligned}(X^T \Omega^{-1} X)^{-1} X^T \Omega^{-1} &= (X^T C^T C X)^{-1} X^T C^T C \\ &= ((CX)^T (CX))^{-1} (CX)^T C \\ &= C.\end{aligned}$$

So Cy is the GLS for covariance matrix $(C^T C)^{-1}$. □

[2] This captures the expected value for the square of the difference between the true value at λ (which is $\beta \cdot \lambda$) and the estimated value at λ (which is $\hat{\beta} \cdot \lambda$).

D.3 Missing proof of Theorem 6

Theorem 12 (Restatement of Theorem 6)**.** *For every $d \times n$ matrix C satisfying $CX = I_d$, the algorithm GWA_C is ε-DP and the total variance $\sum_{j=1}^{d} Var(\tilde{\beta}_j)$ of the vector $\tilde{\beta}$ produced by the algorithm GWA_C can be written as:*

$$\sigma^2 \sum_{j=1}^{d} \sum_{i=1}^{n} c_{j,i}^2 + 2d \left(B\varepsilon^{-1} \cdot \max_l \sum_{j=1}^{d} \sum_{i \in S_l} |c_{j,i}| \right)^2 .$$

This total variance is a convex function and be minimized in polynomial time.

Proof. For privacy guarantee, we analyze the ℓ_1-sensitivity of $\hat{\beta}$. If user l change its input y_i's for $i \in S_l$, $\hat{\beta}$ would be changed by at most $B \cdot \sum_{j=1}^{d} \sum_{i \in S_l} |c_{j,i}|$. Therefore. the ℓ_1-sensitivity of $\hat{\beta}$ is at most $\max_{l=1}^{m} B \cdot \sum_{j=1}^{d} \sum_{i \in S_l} |c_{j,i}|$. By Theorem 1, we know that $\tilde{\beta}$ is ε-DP.

The calculation of the total variance is straightforward and we omitted it here.

For convexity of the total variance, we prove the claim simply by using the following facts sequentially: (1) the absolute value $f(x) = |x|$ is a convex function (2) the max of convex functions is convex (3) the square of a convex non-negative function is convex (4) the sum of convex functions is convex. □

D.4 Missing proof of Claim 3 and a different gap example

Claim 8 (Restatement of Claim 3)**.** *In Example 1, the minimum variance of the sample limiting algorithm is at least $1/4g^3$ and the minimum variance of the generalized weighted averaging algorithm is at most $1/g^4$.*

Proof. For any threshold $h \in [g]$, the sample limiting algorithm has variance

$$\max \left(\frac{g}{g^2 + h \cdot g^2}, \frac{h}{h + g^2} \right)^2 \geq \max \left(\frac{1}{2gh}, \frac{h}{2g^2} \right)^2 \geq \frac{1}{4g^3}.$$

Now we set some C for the generalized weighted averaging algorithm. For data points i from user $2, ..., g^2 + 1$, we set $c_{1,i}$ to be $1/g^2, 0$. For data points i from user $g^2 + 3, ..., 2g^2 + 2$, we set $c_{2,i}$ to be $1/g^2$. We set other entries of C to be zero. With this C, the generalized weighted averaging algorithm has variance

$$0 + 2d \left(\frac{B}{\varepsilon g^2} \right)^2 = \frac{1}{g^4}.$$

□

Example 3. *$d = 2$. Let $g \geq 8$ be some integer parameter and set the number of users to be $m = g+1$. Set $\sigma = 1$ and $2d(B/\varepsilon)^2 = g$. Now consider the data points of users:*

- *User 1 has g data points with $X_i = (1, 0)$.*
- *Each of user $2, ..., g + 1$ has 1 data point with $X_i = (1, 0)$ and $g - 1$ data points with $X_i = (0, 1)$.*

Claim 9. *In Example 3, the minimum variance of the generalized weighted averaging algorithm is at most $6/g$. For any threshold h, with probability at least $1/2$, the sample limiting algorithm has variance at least $g/9$.*

Proof. We set some C for the generalized weighted averaging algorithm. For each data point i with $X_i = (1, 0)$ from user $2, ..., g + 1$, we set $c_{1,i} = 1/g$. For each data point i with $X_i = (0, 1)$ from user $2, ..., g + 1$, we set $c_{2,i} = 1/(g(g - 1))$. We set other entries of C to be zero. With this C, the generalized weighted averaging algorithm has variance

$$1/g + 1/(g(g - 1)) + 2d \left(\frac{2B}{\varepsilon g} \right)^2 \leq \frac{6}{g}.$$

Consider the sample limiting algorithm with threshold h. For each of user $2, ..., g+1$, the probability that the data point with $X_i = (1,0)$ is sampled is at most h/g. Let O be the number of sampled $X_i = (1,0)$ from user $2, ..., g+1$. We have $\mathbb{E}[O] \leq g \cdot \frac{h}{g} = h$. By Markov inequality, we know that with probability at least $1/2$, $O \leq 2h$. In this case, $\sum_{j=1}^{d} \sum_{i \in S_1^s} |c_{j,i}^s| \geq \frac{h}{h+O} \geq 1/3$ and the variance of $\tilde{\beta}^s$ is at least

$$2d(B/\varepsilon)^2 \left(\sum_{j=1}^{d} \sum_{i \in S_1^s} |c_{j,i}^s| \right)^2 \geq g/9.$$

□

E Additional material for the experimental analysis.

E.1 Missing details of linear regression experiments

E.1.1 Detailed description of the datasets

Synthetic data We generated instances of regression problems with multiple user contributions using the `sklearn`'s package `make_regression` method. This method generates a random X, y instance with a fix number d_p (d_n) of predictive (non-predictive) features, as well as a controllable noise. We use $n \in [600, 3000]$ for the number of samples, $d_p = d_n = 5$, we set bias=0.0 and noise=20. To generate the number of row contributions per user we follow approximately the Zipf's (power law) distribution [AH02]. It is well known that users contribute to many systems with heavy-tailed distributions (many users have few contributions, but some have many) and the Zipf's law is well used in the literature [AH02]. We fix the number of users u_i with i rows, to follow the probability mass function of the Zipf's law of parameter $\alpha > 1$ (i.e. $u_i \propto \frac{1}{i^\alpha}$). We use $1 < \alpha \leq 2$ in our experiments. Then users are assigned the prescribed number of rows randomly.

Real-world datasets All real-world datasets are available on the UCI Machine Learning Dataset repository[3].

drugs[4] [GKMZ18] The dataset contains reviews of drugs by anonymous users. Each row corresponds to a drug review. The features correspond to drug characteristics (e.g., side-effects, disease treated, etc). The target to predict is the numeric rating given by the user to the drug (in range $[1, 10]$). As standard in linear regression, categorical features with k values are encoded as a $k-1$ dimensional one-hot encoding dropping one category. We use the drugs as the partitions of the rows in users. We have $n = 3107$ samples, $d = 8$ features, $m = 502$ users with min 1 and max 63 samples.

news[5] [MT18] The dataset contains the popularity of social media posts at different times. Each row corresponds to a post. The features corresponds to the popularity of the post at intervals of 20 minutes for the first 10 hours from publication. The target to predict is the final popularity after 48 hours from publication (the range is $[0, 71]$). We use the news source as the partitions of the rows in users. We have $n = 3452$ samples, $d = 10$ features, $m = 297$ users with min 1 and max 878 samples.

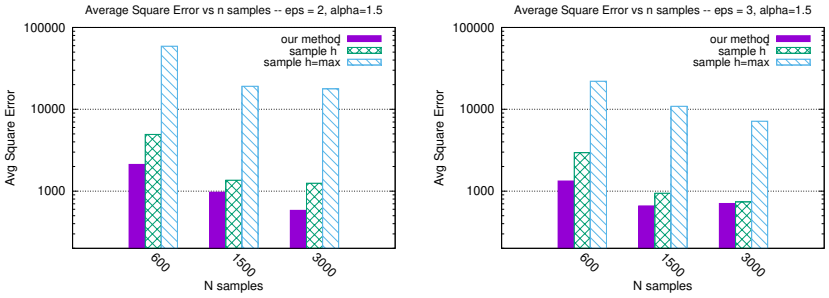
E.1.2 Detailed description of the experimental set up

For each dataset and each setting and parameter setting we run the algorithms 10 times and report mean of each metric computed. For each run of the algorithms, we use as quality measure the empirical average squared error for the prediction. We evaluate our general setting algorithm in Section 5, where the only two parameters are the ε value of differential privacy and the σ^2 variance of the noise in the regression. We evaluate all algorithms using $\varepsilon = 1, 2, 3$ values. For σ^2 we plug in the variance computed by a standard ordinary least squares method (we treat it as public knowledge for simplicity of evaluation). We compare with the baseline of the **sample limiting** algorithm. For this algorithm we compare the results of the best threshold h^* obtained by empirical evaluation (i.e., the one with the lowest mean empirical loss) and the method using all datasets (i.e. h set to max user

[3] `https://archive.ics.uci.edu/ml/datasets.php`

[4] `https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Druglib.com%29`

[5] `https://archive.ics.uci.edu/ml/datasets/News+Popularity+in+Multiple+Social+Media+Platforms`



(a) $\varepsilon = 2$ (b) $\varepsilon = 3$

Figure 3: Average squared errors for our method (solid bar) and the sample limiting method with best thresholds ('X' pattern), as well for the whole sample ('\' pattern), for the synthetic dataset with $\alpha = 1.5$ and various ε parameters. – additional results

contribution). Finally, to solve the constrained optimization problem in our method we use the SCS solver of `CVXPY`[6].

E.1.3 Additional results for the synthetic data

In Figure 3 we report additional experimental results for other settings of ε. The trends observed for $\varepsilon = 1$ are confirmed over the entire range. As expected, with higher ε values, all methods improve their performances, but we still see that our method has lower loss. Notice also the large gap between using the best threshold and the entire dataset for the sample limiting method.

E.2 Experiment results on logistic regression with ERM algorithms

For completeness, although our theory suggests that the weighted averaging algorithm won't outperform the sample limiting algorithm by much for ERM, we perform experiments on logistic regression using our weighted averaging algorithm and compare it to sample limiting. We use cross-entropy loss as the loss function. For a data point with feature x and binary label y, the cross-entropy loss is defined as

$$l(\theta, (x, y)) = y \cdot \ln\left(\frac{1}{1 + e^{-\theta \cdot x}}\right) + (1 - y) \cdot \ln\left(\frac{e^{-\theta \cdot x}}{1 + e^{-\theta \cdot x}}\right).$$

We apply our weighted averaging algorithm (Algorithm 2). By Claim 1, we know the optimal weights can be characterized by a single parameter h, and data points from user j get weight

$$\frac{1}{s_j} \cdot \frac{\min(s_j, h)}{\sum_{l=1}^{m} \min(s_l, h)}.$$

The sample limiting algorithm can also be characterized by a single parameter h (needs to be an integer). For each user j, $\min(s_j, h)$ of its data points will be used.

We use **drugs** dataset mentioned in the linear regression experiments. We make the label binary by splitting them into rating above the median and rating below the median.

We optimize θ over $[-10, 10]^d$ and we clip the gradient of each data point at ℓ_2 norm 1 before taking the weighted sum. For interesting comparison such that the weighted averaging algorithm and the sample limiting algorithm have reasonable performance, we set $\delta = 0.1$ and $\varepsilon = 30, 40, 50$. For each setting, we run the algorithm 50 times. We show in Table 2 the average loss of the settings. For both algorithms, $h \leq 4$ are the interesting region.

As shown in Table 2, the weighted averaging algorithm has small advantage over the sample limiting algorithm. Both the weighted averaging algorithm and the sample limiting algorithm outperform the algorithm which does not bound user contribution.

[6] `https://www.cvxpy.org/tutorial/advanced/index.html`

Dataset	ε	**Weighted averaging**				**Sample limiting**				**No bounds on user contributions**
		h=1	h=2	h=3	h=4	h=1	h=2	h=3	h=4	
drugs	30	0.514	0.557	0.628	0.700	0.553	0.580	0.654	0.725	2.846
	40	0.433	0.469	0.510	0.551	0.474	0.472	0.522	0.553	2.611
	50	0.401	0.416	0.445	0.472	0.436	0.423	0.454	0.464	2.307

Table 2: Average loss for logistic regression.