sciendo

Proceedings on Privacy Enhancing Technologies ; 2022 (2):184–204

Daniel Alabi*, Audra McMillan, Jayshree Sarathy, Adam Smith, and Salil Vadhan

Differentially Private Simple Linear Regression

Abstract: Economics and social science research often require analyzing datasets of sensitive personal information at fine granularity, with models fit to small subsets of the data. Unfortunately, such fine-grained analysis can easily reveal sensitive individual information. We study regression algorithms that satisfy *differential privacy*, a constraint which guarantees that an algorithm's output reveals little about any individual input data record, even to an attacker with side information about the dataset. Motivated by the *Opportunity Atlas*, a high-profile, small-area analysis tool in economics research, we perform a thorough experimental evaluation of differentially private algorithms for simple linear regression on small datasets with tens to hundreds of records—a particularly challenging regime for differential privacy. In contrast, prior work on differentially private linear regression focused on multivariate linear regression on large datasets or asymptotic analysis. Through a range of experiments, we identify key factors that affect the relative performance of the algorithms. We find that algorithms based on robust estimators—in particular, the median-based estimator of Theil and Sen—perform best on small datasets (e.g., hundreds of datapoints), while algorithms based on Ordinary Least Squares or Gradient Descent perform better for large datasets. However, we also discuss regimes in which this general finding does not hold. Notably, the differentially private analogues of Theil–Sen (one of which was suggested in a theoretical work of Dwork and Lei) have not been studied in any prior experimental work on differentially private linear regression.

Keywords: differential privacy, linear regression, robust statistics

DOI 10.2478/popets-2022-0041
Received 2021-08-31; revised 2021-12-15; accepted 2021-12-16.

***Corresponding Author: Daniel Alabi:** Harvard John A. Paulson School of Engineering and Applied Sciences, E-mail: alabid@g.harvard.edu
Audra McMillan: Khoury College of Computer Sciences, Northeastern University and Department of Computer Science, Boston University, Email: audramarymcmillan@gmail.com
Jayshree Sarathy: Harvard John A. Paulson School of Engineering and Applied Sciences, E-mail: jsarathy@g.harvard.edu
Adam Smith: Department of Computer Science, Boston University, E-mail: ads22@bu.edu

1 Introduction

The analysis of small datasets, with sizes in the dozens to low hundreds of records, is crucial in many social science applications. For example, neighborhood-level household income, high-school graduation rate, and incarceration rates are all studied using sensitive datasets that are subdivided into small, local units to allow for fine-grained inspection (e.g., [12]). As datasets get larger, they are subdivided more finely. Handling small samples automatically and reliably is therefore crucial even for working with big data. However, the release of statistical estimates based on these data quantities—if too many and too accurate—can allow reconstruction of the original dataset [17]. The possibility of such attacks led to differential privacy [19], a rigorous mathematical definition used to quantify privacy loss. Differentially private (DP) algorithms limit the information that is leaked about any particular individual by introducing random distortion. The amount of distortion, and its effect on utility, are most often studied for large datasets, using asymptotic tools. When datasets are small, one has to be very careful when calibrating differentially private statistical estimates to preserve utility.

In this work, we focus on the prominent statistical task of simple (i.e., one-dimensional) linear regression, which is a workhorse of analysis in many social science fields [16]. Our goal is to provide methodology for performing differentially private linear regression in practical applications. In particular, we are motivated by the small-area regressions that underpin the high-profile Opportunity Atlas [13]. We show that differentially private linear regression can be accurate even on small datasets. We will provide insight and guidance into how to choose a DP algorithm for simple linear regression in a variety of realistic parameter regimes. Our work differs from previous work on differentially private linear regression in its emphasis on small datasets (previous works mostly focus on asymptotic theoretical analysis) and in our evaluation on a real application.

Salil Vadhan: Harvard John A. Paulson School of Engineering and Applied Sciences, E-mail: salil_vadhan@harvard.edu

Even without a privacy constraint, small sample sizes pose a problem for statistical inference, since the variability from sample to sample, called the *sampling error*, can overwhelm the signal about the underlying trend. A reasonable concrete goal for a DP mechanism, then, is that it not introduce substantially *more* uncertainty into the estimate than the sampling error. Specifically, we compare the noise added in order to maintain privacy to the standard error of the nonprivate estimate, obtained using Ordinary Least Squares (OLS). Our experiments indicate that for a wide range of realistic datasets and moderate values of the privacy parameter, ε, it is possible to choose a DP linear regression algorithm that introduces distortion less than the standard error. In particular, in our motivating use-case, the Opportunity Atlas [13], we provide a differentially private algorithm, `DPTheilSen`, that matches or outperforms the heuristic method currently deployed, which does not formally satisfy differential privacy.

We focus on univariate linear regression because, as shown by the Opportunity Atlas deployment, choosing the right algorithm for this basic task when data is limited is a challenge. This problem remains a barrier to adoption of differential privacy for many practical applications. While the univariate case may seem much simpler than the higher dimensional cases studied in many previous works, the algorithms that we show are commonly the best performing in the small dataset regime, `DPTheilSen` and its variants, were not considered in prior experimental work, including in the systematic empirical evaluation conducted by Wang [29]. Prior work focuses almost exclusively on the large dataset regime, and our work shows that there is a fundamental shift in the types of algorithms that should be considered in the small versus large dataset regimes. One of the variants of Theil-Sen we consider, `DPTheilSen1Match`, was proposed by Dwork and Lei [18], who gave a theoretical, asymptotic analysis of its accuracy. We are not aware of any prior experimental evaluation of any of the `DPTheilSen` variants.

2 Preliminaries

2.1 Differential Privacy

The algorithms in this paper satisfy differential privacy (DP). Since our algorithms often include hyperparameters, we state a definition of DP for algorithms that take as input not only the dataset, but also the desired privacy parameters and any required hyperparameters. Let $\mathcal{X}$ be a data universe (e.g., $\mathbb{R}^2$ for simple linear regression) and $\mathcal{X}^n$ be the space of datasets. Two datasets $d, d' \in \mathcal{X}^n$ are neighboring, denoted $d \sim d'$, if they differ on a single record. Let $\mathcal{H}$ be a hyperparameter space and $\mathcal{Y}$ be an output space.

Definition 1 ((ε, δ)-Differential Privacy [19])**.** *A randomized mechanism $M : \mathcal{X}^n \times \mathbb{R}_{\geq 0} \times [0,1] \times \mathcal{H} \to \mathcal{Y}$ is* differentially private *if for all datasets $d \sim d' \in \mathcal{X}^n$, privacy-loss parameters $\varepsilon \geq 0, \delta \in [0,1]$, $hp \in \mathcal{H}$, and events $E \subseteq \mathcal{Y}$,*

$$\Pr[M(d, \varepsilon, \delta, hp) \in \mathrm{E}] \leq e^{\varepsilon} \cdot \Pr[M(d', \varepsilon, \delta, hp) \in \mathrm{E}] + \delta,$$

where probabilities are taken over M's random coins.

For strong privacy guarantees, the privacy-loss parameter is typically taken to be a small constant less than 1 (note that $e^{\varepsilon} \approx 1 + \varepsilon$ as $\varepsilon \to 0$), but we will sometimes consider larger constants such as $\varepsilon = 8$ to match what was used in our motivating application (described in Section 3).

2.2 Simple Linear Regression

In this paper, we consider the most common model of linear regression: we are given n observations $x_1, \ldots, x_n \in \mathbb{R}$ of an explanatory variable and corresponding observations $y_1, \ldots, y_n \in \mathbb{R}$ of a response variable. We wish to find a slope $\alpha \in \mathbb{R}$ and an intercept $\beta \in \mathbb{R}$ such that, y_i is well-approximated by $\alpha \cdot x_i + \beta$. Specifically, we consider the Ordinary Least Squares (OLS) objective characterized by the following optimization problem:

$$(\hat{\alpha}, \hat{\beta}) = \arg \min_{\alpha, \beta \in \mathbb{R}} \|\mathbf{y} - \alpha \mathbf{x} - \beta \mathbf{1}\|_2, \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)^T$, $\mathbf{y} = (y_1, \ldots, y_n)^T$, and $\mathbf{1}$ is the all-ones vector. This is the most commonly used linear regression formulation in practice. Indeed, when $\mathbf{y}$ is generated according to the model $y_i = \alpha \cdot x_i + \beta + e_i, \forall i \in [n]$ for i.i.d. Gaussian noise e_i, then the OLS solution is the maximum likelihood estimator for the "ground truth" parameters α, β. Moreover, OLS has a simple closed form solution:

$$\hat{\alpha} = \frac{n\mathrm{cov}(\mathbf{x}, \mathbf{y})}{n\mathrm{var}(\mathbf{x})} \text{ and } \hat{\beta} = \bar{y} - \hat{\alpha}\bar{x}, \tag{2}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^n x_i$, $\bar{y} = \frac{1}{n}\sum_{i=1}^n y_i$, $n\mathrm{cov}(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x} - \bar{x}\mathbf{1}, \mathbf{y} - \bar{y}\mathbf{1} \rangle$, and $n\mathrm{var}(\mathbf{x}) = \langle \mathbf{x} - \bar{x}\mathbf{1}, \mathbf{x} - \bar{x}\mathbf{1} \rangle$.

In this paper, we focus on predicting the (mean of the) response variable y at a single value of the explanatory variable x. For $x_{new} \in \mathbb{R}$, the *prediction* at x_{new} is defined as:

$$p_{x_{new}} = \alpha x_{new} + \beta$$

Let $\widehat{p}_{x_{new}}$ be the prediction at x_{new} computed using the OLS estimates $\hat{\alpha}$ and $\hat{\beta}$. The corresponding DP estimates will be denoted by $\widetilde{p}_{x_{new}}$. The quantities $\widehat{p}_{x_{new}}$ and $\widetilde{p}_{x_{new}}$ are random variables, where the randomness is due to the sampling process and/or the noise added to ensure privacy. After normalizing the independent variable to lie in the interval $[0, 1]$, we will be primarily concerned with the predicted values at $x_{new} = 0.25$ and 0.75, which for ease of notation we denote as p_{25} and p_{75}, respectively. Correspondingly, we will use $\widehat{p}_{25}$, $\widehat{p}_{75}$ to denote the OLS estimates of the predicted values and $\widetilde{p}_{25}$, $\widetilde{p}_{75}$ to denote the DP estimates. Note that computing both of these predicted values allows one to derive the full simple linear regression model. In this paper, we focus on the error on the predicted values rather than the slope itself in order to compare directly with the Opportunity Atlas tool [12], described in Section 3, which releases estimates of p_{25} and p_{75} for certain regressions done for every census tract in each state. However, we also discuss how the error on these predicted values relates to the error on the slope.

2.3 Error Metric

In measuring the performance of different algorithms, we will focus on high probability error bounds that can be accurately estimated through Monte Carlo experiments. Providing tight theoretical error bounds for DP linear regression is an important direction for future work. Since the relationship between the OLS estimate $\widehat{p}_{x_{new}}$ and the true value $p_{x_{new}}$ is well-understood, we focus on measuring the difference between the private estimate $\widetilde{p}_{x_{new}}$ and $\widehat{p}_{x_{new}}$ Specifically, we define the *prediction error* at x_{new} to be $|\widetilde{p}_{x_{new}} - \widehat{p}_{x_{new}}|$.

Note that the slope α can be computed as $(\mathrm{p}_{75} - \mathrm{p}_{25})/(0.75 - 0.25)$, and thus if we estimate the slope $\tilde{\alpha}$ using differentially private estimates $\widetilde{p}_{75}$ and $\widetilde{p}_{25}$, the error (compared to the OLS estimate $\hat{\alpha}$) will be at most a constant factor larger than the prediction errors at .25 and .75:

$$|\tilde{\alpha} - \hat{\alpha}| \leq 2 \cdot (|\widetilde{p}_{75} - \widehat{p}_{75}| + |\widetilde{p}_{25} - \widehat{p}_{25}|).$$

Furthermore, in Section 7.3.2, we also experimentally evaluate the error on the slope and find that the comparison between algorithms is very similar to what we see for the point estimate errors.

For a dataset $d \in \mathcal{X}^n$, $x_{new} \in \mathbb{R}$, and $q \in [0, 1]$, we define the *q error bound* as

$$C(q)(d) = \min\left\{c : \mathbb{P}(|\widetilde{p}_{x_{new}} - \widehat{p}_{x_{new}}| \leq c) \geq q\right\},$$

where the dataset d is fixed, and the probability is taken over the randomness in the DP algorithm.

We empirically estimate $C(q)$ by running many trials of the algorithm on the same dataset d:

$$\hat{C}(q)(d) = \min\{c : \text{for at least } q \text{ fraction of trials, } |\widetilde{p}_{x_{new}} - \widehat{p}_{x_{new}}| \leq c\}.$$

We term $\hat{C}(q)(d)$, the *q empirical error bound*. We will often drop the reference to d from the notation, but note that the error metric $|\widetilde{p}_{x_{new}} - \widehat{p}_{x_{new}}|$ only accounts for the randomness in the algorithm, not the sampling error.

When the ground truth slope α and intercept β are known (e.g., for synthetically generated data), we can compute error bounds compared to the ground truth, rather than to the non-private OLS estimate. So, let $C_{\text{true}}(q)(d)$ and $\hat{C}_{\text{true}}(q)(d)$ be similar to the error bounds described earlier, except that the prediction error is measured as $|\widetilde{p}_{x_{new}} - p_{x_{new}}|$. This error metric accounts for the randomness in both the sampling and the algorithm.

The standard error $\widehat{\sigma}(\widehat{p}_{x_{new}})$ is an estimate of the standard deviation of $\widehat{p}_{x_{new}}$, $\sigma(\widehat{p}_{x_{new}})$. We use the following formula, which is an unbiased estimate of $\sigma(\widehat{p}_{x_{new}})$ in the case where $y_i = \alpha \cdot x_i + \beta + e_i$ for all $i \in [n]$ and for i.i.d. Gaussian e_i (see [30], for example):

$$\widehat{\sigma}(\widehat{p}_{x_{new}}) = \frac{\|\mathbf{y} - \hat{\alpha}\mathbf{x} - \hat{\beta}\|_2}{\sqrt{n-2}} \sqrt{\frac{1}{n} + \frac{(x_{new} - \bar{x})^2}{n\mathrm{var}(\mathbf{x})}}. \quad (3)$$

Even for non-Gaussian e_i, it can be shown that the variance of $(\widehat{p}_{x_{new}} - p_{x_{new}})/\widehat{\sigma}(\widehat{p}_{x_{new}})$ approaches 1 as n increases.

When we say *the noise added for privacy is less than the sampling error*, we are referring to the technical statement that $\hat{C}(0.68)$ is less than the standard error, $\widehat{\sigma}(\widehat{p}_{x_{new}})$. [1] We stress that the methods we develop do *not* require that the noise distribution be Gaussian; the noise model assumption is only used to derive the formula for the non-private standard deviation (Equation 3) that we use as a benchmark to compare against.

1 Note that without privacy and under an assumption of normally distributed noise, we expect $\hat{C}(0.68)$ to be roughly equal to $\hat{\sigma}(\widehat{p}_{x_{new}})$.

3 Motivating Use-Case: Opportunity Atlas

The Opportunity Atlas, designed and deployed by the economics research group Opportunity Insights, is an interactive tool designed to study the link between the neighbourhood a child grows up in and their prospect for economic mobility [12]. The tool provides valuable insights to researchers and policy-makers, and has received much press since its release (e.g., see coverage by the New York Times [5]). It is built by linking two data sources: Census data, protected under Title 13 and authorized by the Census Bureau's Disclosure Review Board, and federal income tax returns from the US Internal Revenue Service. The Atlas provides individual statistics on each census tract in the country, with tract data often being refined by demographics to contain only a small subset of the individuals who live in that tract. The resulting datasets typically contain 100 to 400 datapoints, but can be as small as 30 datapoints. The response variable $y_i \in [0,1]$ is the child's income percentile at age 35 and the explanatory variable $x_i \in [0,1]$ is the parent's income percentile, each with respect to the national income distribution. The coefficient α in the model $y_i = \alpha \cdot x_i + \beta + e_i$ is a measure of economic mobility for that particular Census tract and demographic. The Atlas releases separate estimates of p_{25} and p_{75} (the predicted values at $x_{new} = 0.25$ and 0.75) for each census tract. The small size of the datasets used in the Opportunity Atlas are the result of Chetty et al.'s desire to study inequality at the neighbourhood level. This fine granularity is crucial to the predictive power of the tool, and for providing insight for local policy makers. According to Chetty et al., "the estimates permit precise targeting of policies to improve economic opportunity by uncovering specific neighborhoods where certain subgroups of children grow up to have poor outcomes. Neighborhoods matter at a very granular level: conditional on characteristics such as poverty rates in a child's own Census tract, characteristics of tracts that are one mile away have little predictive power for a child's outcomes" [12].

This type of highly local and fine-grained analysis is typical of social sciences, where one often wants to subdivide the population as finely as possible, subject to the units having enough size for some particular statistical analysis. Working at that sample size limit poses privacy risks, which makes it a challenging but important regime for DP.

4 Robustness and DP Algorithm Design

Simple linear regression is one of the most fundamental statistical tasks with well-understood convergence properties in the non-private literature. However, finding a differentially private estimator for this task that is accurate across a range of datasets and parameter regimes is surprisingly nuanced.

While there has been a significant amount of prior work on differentially private linear regression, often in even more general settings than we consider (such as multivariate regression or even general convex optimization), our work appears to be unique in its focus on achieving high utility on small datasets. Indeed, prior work has focused on asymptotic performance (as the sample size n grows), giving either theoretical bounds with large or unspecified constants (e.g., [6, 11, 18, 25, 26]) or experimental evaluation on datasets of size at least $n = 1,000$ (e.g., [25, 29][2]). This type of analysis makes it difficult to compare the relative performance of these algorithms on small datasets. In contrast, we provide a thorough experimental evaluation on datasets of size in the range $n = 30$ to 400 datapoints for the Opportunity Atlas use-case and $n = 30$ to $10,000$ on synthetically generated data. This regime requires a fundamental change in the types of algorithms that are considered.

As a first attempt to construct a differentially private estimator for this task, one might consider the global sensitivity [19]:

Definition 2 (Global Sensitivity)**.** *For a query $f : \mathcal{X}^n \to \mathbb{R}^k$, the **global sensitivity** is*

$$GS_f = \max_{d \sim d'} \|f(d) - f(d')\|_1.$$

One can create a differentially private mechanism by adding noise proportional to GS_f/ε. Unfortunately, the global sensitivity of $\widehat{p}_{25}$ and $\widehat{p}_{75}$ are both infinite (even if we clip each (x, y) datapoint to lie within a bounded range, like in the Opportunity Atlas use-case, the point estimates $\widehat{p}_{25}$ and $\widehat{p}_{75}$ are unbounded). For the type of datasets that we typically see in practice, however, changing one datapoint does not result in a major change in the point estimates. For such datasets, where

2 Wang [29] conducts one experiment with $n = 506$, but the rest of the experiments use tens of thousands of datapoints.

the point estimates are reasonably stable, one might hope to take advantage of the local sensitivity:

Definition 3 (Local Sensitivity [23])**.** *The **local sensitivity** of a query $f : \mathcal{X}^n \to \mathbb{R}^k$ with respect to a dataset $d \in \mathcal{X}^n$ is*

$$LS_f(d) = \max_{d \sim d'} \|f(d) - f(d')\|_1.$$

Unfortunately, adding noise proportional to the local sensitivity is typically **not** differentially private, since the local sensitivity itself can reveal information about the underlying dataset $d \in \mathcal{X}^n$. There are several approaches in the DP literature that instead add noise proportional to an appropriate upper bound on the local sensitivity, such as through the *smooth sensitivity* framework [23], a DP high-probability upper bound on the local sensitivity [18], or more recent variants. It is an interesting open problem to design a smooth sensitivity algorithm for linear regression or to derive DP upper bounds on the local sensitivity of linear regression that are not too loose for our setting.

The Opportunity Insights (OI) algorithm takes a heuristic approach by adding noise proportional to a **non-private**, heuristic upper bound on the local sensitivity of data from tracts in any given state. However, their heuristic approach does not satisfy the formal requirements of differential privacy, leaving open the possibility that there exists a realistic attack.

The OI algorithm incorporates a "winsorization" step in their estimation procedure (e.g., dropping the top and bottom 10% of data values). This sometimes has the effect of greatly reducing the local sensitivity (and also their upper bound on it) due to the possible removal of outliers. This suggests that for finding an effective differentially private algorithm, we should consider differentially private analogues of *robust* linear regression methods rather than of OLS. Specifically, we consider *Theil-Sen*, a robust estimator for linear regression proposed by Theil [28] and further developed by Sen [24]. Similar to the way in which the median is more robust to changes in the data than the mean, the Theil-Sen estimator is more robust to changes in the data than OLS .

Motivated by the above conditions, we consider three differentially private algorithms based on both robust and non-robust methods:

`DPSuffStats` is the DP mechanism that most closely mirrors OLS. It involves perturbing the sufficient statistics $n\text{cov}(\mathbf{x}, \mathbf{y})$ and $n\text{var}(\mathbf{x})$ from the OLS computation. This algorithm is related to the "Analyze Gauss" technique [20], which is the basis of some of the algorithms considered by [25] and [29]. However, we deviate from prior work in using Laplace noise, rather than Gaussian noise, to ensure pure differential privacy rather than approximate differential privacy. `DPSuffStats` has two main benefits: it is as computationally efficient as its non-private analogue, and it allows us to release DP versions of the sufficient statistics with no extra privacy cost.

`DPTheilSen` is a DP version of Theil-Sen. The non-private estimator computes the p_{25} estimates based on the lines defined by all pairs of datapoints $(x_i, y_i), (x_j, y_j)$ for all $i \neq j \in [n]$, then outputs the median of these pairwise estimates. To create a differentially private version, we replace the median computation with a differentially private median algorithm. We consider three DP versions of this algorithm which use different DP median algorithms: `DPExpTheilSen`, `DPWideTheilSen`, and `DPSSTheilSen`. We also consider more computationally efficient variants that pair points according to one or more random matchings, rather than using all $\binom{n}{2}$ pairs. A DP algorithm obtained by using one matching was previously considered by Dwork and Lei [18] (their "Short-Cut Regression Algorithm"). Our algorithms can be viewed as updated versions of their approach, reflecting improvements in DP median estimation since [18], as well as incorporating benefits accrued by considering more than one matching.

`DPGradDescent` is a DP mechanism that uses DP gradient descent to solve the convex optimization problem that defines OLS: $\text{argmin}_{\alpha,\beta} \|\mathbf{y} - \alpha\mathbf{x} - \beta\mathbf{1}\|_2$. We use the private stochastic gradient descent technique proposed by [6]. Versions that satisfy pure, approximate, and zero-concentrated differential privacy are considered.

While a variant of `DPTheilSen` does appear in [18], to the best of our knowledge it does not appear in any experimental studies exploring differentially private algorithms for linear regression. We find that `DPTheilSen` is particularly effective in the small dataset regime, and can significantly outperform other DP algorithms in this regime.

5 Related Work

Linear regression is one of the most prevalent statistical methods in the social sciences, and hence has been studied previously in the differential privacy literature. These works have included both theoretical analysis and

experimental exploration, with the majority of work focusing on large datasets.

One of our main findings — that robust estimators perform better than parametric estimators in the differentially private setting, even when the data come from a parametric model — corroborate insights by Dwork and Lei [18] with regard to the connection between robust statistics and differential privacy, and by Couch et al. [15] in the context of hypothesis testing.

As discussed earlier, systematic studies of DP linear regression have been performed in several prior works. Sheffet [25] considered differentially private ordinary least squares methods and estimated confidence intervals for the regression coefficients. When we plug in parameters from our experiments[3], Sheffet's main algorithm (OLS over projected data) requires n to be larger than 431 at a minimum. Sheffet runs experiments on synthetic datasets with sizes ranging from $n = 10^3$ to $n = 10^5$, and on real datasets with sizes ranging from $n = 30,000$ to $n = 70,000$. Sheffet's algorithms perform well (i.e., they correctly reject the null hypothesis with high probability) only once the dataset size n is at least $10,000$. Wang [29] considered private ridge regression, using techniques similar to output perturbation [11]. Wang runs some experiments on synthetic data with n ranging from 50 to 10^7 datapoints, but the real dataset evaluations included are focused on large datasets with n ranging from 500 to $583,250$ datapoints. These previous works on DP linear regression present methods and experiments for multi-dimensional data (e.g., the datasets used in [29] contain at least 13 explanatory variables), whereas we are concerned with the one-dimensional setting and the small dataset regime.

A Bayesian approach to DP linear regression is taken by Bernstein and Sheldon [7] which, unlike ours, requires a prior on the distribution of both the regression coefficients and the independent variables. Zhang et al. [31] and Awan and Slavkovic [4] study the functional mechanism for linear regression, and Cai et al. [10] provide sharp minimax bounds for linear regression under (ε, δ)-differential privacy constraints. But these works do not pertain to the small dataset one-dimensional regime.

In all of the works mentioned above, the theoretical utility guarantees are asymptotic, and the experimental results are focused on large datasets (i.e., with tens of thousands of datapoints), so it is difficult to ascertain the utility in the small dataset regime. None of the previous experimental works have considered `DPTheilSen` and its variants, which we find to be the best performing set of algorithms in this challenging regime.

6 Algorithms

In this section we detail the practical differentially private algorithms we will evaluate experimentally. Pseudocode for all efficient implementations of each algorithm described can be found here or later on in the Appendix, and real code can be found in our GitHub repository: https://github.com/anonymous-conf/dplr.

6.1 `DPSuffStats`

In `DPSuffStats` (Algorithm 1), we add Laplace noise, with standard deviation approximately $1/\varepsilon$, to the OLS sufficient statistics, $n\text{cov}(\mathbf{x}, \mathbf{y}), n\text{var}(\mathbf{x})$, and then use the noisy sufficient statistics to compute the predicted values. Note that this algorithm fails if the denominator for the OLS estimator, the noisy version of $n\text{var}(\mathbf{x})$, becomes 0 or negative, in which case we output $\perp$ (failure). The probability of failure decreases as ε or $n\text{var}(x)$ increases. `DPSuffStats` is the simplest and most efficient algorithm that we will study. In addition, the privacy guarantee is maintained even if we additionally release the noisy statistics $n\text{var}(\mathbf{x}) + L_1$ and $n\text{cov}(\mathbf{x}, \mathbf{y}) + L_2$, which may be of independent interest to researchers. We also note that the algorithm is biased due to dividing by a Laplacian distribution centered at $n\text{var}(\mathbf{x})$.

Lemma 4. *For* $0 \leq r_l \leq r_u$, *Algorithm 1* (`DPSuffStats`) *is* $(\varepsilon, 0)$*-DP.*

6.2 DP TheilSen

The non-private Theil-Sen estimator is a *robust* estimator for linear regression. It computes the p_{25} estimates based on the lines defined by all pairs of datapoints $(x_i, y_i), (x_j, y_j)$ for all $i \neq j \in [n]$, then outputs the median of these pairwise estimates. To create a differentially private version, we can replace the median computation with a differentially private median algorithm. We implement this approach using three DP median algorithms; two based on the exponential mechanism [21] and one based on the smooth sensitivity of [23] and the noise distributions of [9].

3 We set $\sigma_x = 0.08, B = 1, \varepsilon = 1, \delta = 10^{-6}$, and $r = n$.

Algorithm 1: DPSuffStats: $(\varepsilon, 0)$-DP Algorithm

Data: $\{(x_i, y_i)\}_{i=1}^n \in (\mathbb{R} \times \mathbb{R})^n$
Privacy params: ε
Hyperparams: r_l, r_u
For all $i \in [n]$, clip each x_i, y_i to $[r_l, r_u]$
Define $\Delta_1 = \Delta_2 = r_u^2 \cdot (1 - 1/n)$
Sample $L_1 \sim \text{Lap}(0, 3\Delta_1/\varepsilon)$
Sample $L_2 \sim \text{Lap}(0, 3\Delta_2/\varepsilon)$
if $nvar(\boldsymbol{x}) + L_2 > 0$ **then**
  $\tilde{\alpha} = \frac{n\text{cov}(\mathbf{x},\mathbf{y}) + L_1}{n\text{var}(\mathbf{x}) + L_2}$
  $\Delta_3 = r_u/n \cdot (1 + |\tilde{\alpha}|)$
  Sample $L_3 \sim \text{Lap}(0, 3\Delta_3/\varepsilon)$
  $\tilde{\beta} = (\bar{y} - \tilde{\alpha}\bar{x}) + L_3$
  $\widetilde{p}_{25} = 0.25 \cdot \tilde{\alpha} + \tilde{\beta}$
  $\widetilde{p}_{75} = 0.75 \cdot \tilde{\alpha} + \tilde{\beta}$
  return $\widetilde{p}_{25}, \widetilde{p}_{75}$
else
  return $\perp$

In the "complete" version of Theil-Sen, all pairwise estimates are included in the final median computation. A similar algorithm can be run on the point estimates computed using k random matchings of the (x_i, y_i) pairs. The case $k = 1$ amounts to the differentially private "Short-cut Regression Algorithm" proposed by Dwork and Lei [18]. This results in a more computationally efficient algorithm.

We will focus mainly on $k = n - 1$, which we will refer to simply as DPTheilSen and $k = 1$, which we will refer to as DPTheilSenMatch. For any other k, we denote the algorithm by DPTheilSenkMatch. In the following subsections we discuss the different differentially private median algorithms we use as subroutines. The pseudocode for DPTheilSenkMatch is found in Algorithm 2.

The lemma below relates the privacy guarantee of Algorithm 2 to the privacy guarantee of the DP median sub-algorithm, DPmed.

Lemma 5. *If DPmed*$(\mathbf{z}^{(\text{p}_{25})}, \varepsilon, (n, k, hp) = \mathcal{M}(\mathbf{z}^{(\text{p}_{25})}, hp))$ *for some* $(\varepsilon/k, 0)$*-DP mechanism* $\mathcal{M}$*, then Algorithm 2 (*DPTheilSenkMatch*) is* $(\varepsilon, 0)$*-DP.*

Algorithm 2: DPTheilSenkMatch: $(\varepsilon, 0)$-DP Algorithm

Data: $\{(x_i, y_i)\}_{i=1}^n \in (\mathbb{R} \times \mathbb{R})^n$
Privacy params: ε
Hyperparams: n, k, DPmed, hp
$\mathbf{z}^{(\text{p}_{25})}, \mathbf{z}^{(\text{p}_{75})} = [\,]$
Let $\tau_1, \cdots, \tau_{n-1}$ be $n-1$ maximal matchings of K_n, the complete graph on n vertices.[4] Each τ_h is a vector of $\lfloor n/2 \rfloor$ pairs corresponding to edges. There are many possible choices for $\tau_1, \cdots, \tau_{n-1}$; one example is for each $h \in [n-1]$, let π_h be an independent random permutation of $[n-1]$. Then, for $i \in \lfloor n/2 \rfloor$, let $\tau_h[i][0] = \pi(i)$ and $\tau_h[i][1] = \pi(i + \lfloor n/2 \rfloor)$.
for k *iterations* **do**
  Sample (without replacement) $h \in [n-1]$
  for $0 \le i < \lfloor n/2 \rfloor$ **do**
    $j = \tau_h[i][0]$
    $l = \tau_h[i][1]$
    if $(x_l - x_j \neq 0)$ **then**
      $s = (y_l - y_j)/(x_l - x_j)$
      $z_{j,l}^{(p25)} = s\left(0.25 - \frac{x_l + x_j}{2}\right) + \frac{y_l + y_j}{2}$
      $z_{j,l}^{(p75)} = s\left(0.75 - \frac{x_l + x_j}{2}\right) + \frac{y_l + y_j}{2}$
      Append $z_{j,l}^{(p25)}$ to $\mathbf{z}^{(\text{p}_{25})}$ and $z_{j,l}^{(p75)}$ to $\mathbf{z}^{(\text{p}_{75})}$
$\tilde{p}_{25} = \text{DPmed}\left(\mathbf{z}^{(\text{p}_{25})}, \varepsilon/2, (n, k, \text{hp})\right)$
$\tilde{p}_{75} = \text{DPmed}\left(\mathbf{z}^{(\text{p}_{75})}, \varepsilon/2, (n, k, \text{hp})\right)$
return $\tilde{p}_{25}, \tilde{p}_{75}$

6.2.1 DP Median Using Exponential Mechanism

The first differentially private algorithm for the median that we will consider is an instantiation of the exponential mechanism [21], a differentially private algorithm designed for general optimization problems. The exponential mechanism is defined with respect to a utility function u, which maps (dataset, output) pairs to real values. For a dataset $\mathbf{z}$, the mechanism aims to output a value r that maximizes $u(\mathbf{z}, r)$.

Definition 6 (Exponential Mechanism [21])**.** *Given dataset* $\boldsymbol{z} \in \mathbb{R}^n$ *and the range of the outputs,* $[r_l, r_u]$*, the exponential mechanism outputs* $r \in [r_l, r_u]$ *with probability proportional to* $\exp\left(\frac{\varepsilon u(\boldsymbol{z}, r)}{2GS_u}\right)$*, where*

$$GS_u = \max_{r \in [r_l, r_u]} \max_{\boldsymbol{z}, \boldsymbol{z}' \, neighbors} |u(\boldsymbol{z}, r) - u(\boldsymbol{z}', r)|.$$

4 A maximal matching here is a subset of the edges of K_n such that no two edges share a vertex and no edge can be added.

One way to instantiate the exponential mechanism to compute the median is by using the following utility function. Let

$$u(\mathbf{z}, r) = -|\#\text{above } r - \#\text{below } r|$$

where #above r and #below r denote the number of datapoints in $\mathbf{z}$ that are above and below r in value respectively, not including r itself. An example of the shape of the output distribution of this algorithm is given in Figure 1. An efficient implementation is given in the Appendix. We will write `DPExpTheilSenkMatch` to refer to `DPTheilSenkMatch` where the DP median function is the DP exponential mechanism described above with privacy parameter ε/k. Again, we write `DPExpTheilSenMatch` when $k = 1$ and `DPExpTheilSen` when $k = n - 1$.

Lemma 7. `DPExpTheilSenkMatch` *is* $(\varepsilon, 0)$*-DP.*

6.2.2 DP Median Using Widened Exponential Mechanism

When the output space is the real line, the standard exponential mechanism for the median has some nuanced behaviour when the data is highly concentrated. For example, imagine in Figure 1 if all the datapoints coincided. In this instance, `DPExpTheilSen` is simply the uniform distribution on $[r_l, r_u]$, despite the fact that the median of the dataset is very stable. To mitigate this issue, we use a variation on the standard utility function. (Concurrently to our work, Asi and Duchi [3] considered a similar utility function, for a special case of their "inverse sensitivity mechanisms.")

For a widening parameter $\theta > 0$, the widened utility function is $u(\mathbf{z}, r) = -\min\{|\#\text{above } a - \#\text{below } a| : |a - r| \leq \theta\}$, where #above a and #below a are defined as before. This has the effect of increasing the probability mass around the median, as shown in Figure 2.

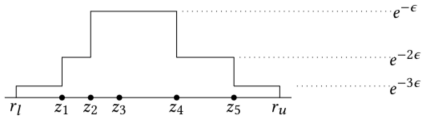


Fig. 1. Unnormalized distribution of outputs of the exponential mechanism for differentially privately computing the median of dataset $\mathbf{z}$.

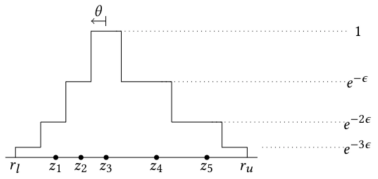


Fig. 2. Unnormalized distribution of outputs of the θ-widened exponential mechanism for differentially privately computing the median of dataset $\mathbf{z}$.

The parameter θ needs to be carefully chosen. All outputs within θ of the median are given the same utility score, so θ represents a lower bound on the expected error. Conversely, choosing θ too small may result in the area around the median not being given sufficient weight in the sampled distribution. We defer the question of optimally choosing θ to future work.

An efficient implementation of the θ-widened exponential mechanism for the median can be found in the Appendix. We will use `DPWideTheilSenkMatch` to refer to `DPTheilSenkMatch` where the DP median mechanism is the θ-widened exponential mechanism with privacy parameter ε/k. Again, we use `DPWideTheilSenMatch` when $k = 1$ and `DPWideTheilSen` when $k = n - 1$.

Lemma 8. `DPWideTheilSenkMatch` *is* $(\varepsilon, 0)$*-DP.*

6.2.3 DP Median Using Smooth Sensitivity Noise Addition

The final algorithm we consider for releasing a differentially private median adds noise scaled to the smooth sensitivity – a smooth upper bound on the local sensitivity function. Intuitively, this algorithm should perform well when the datapoints are clustered around the median; that is, when the median is very stable.

Definition 9 (Smooth Upper Bound on LS_f [23])**.** *For $t > 0$, a function $S_{f,t} : \mathcal{X}^n \to \mathbb{R}$ is a t-*smooth upper bound on the local sensitivity *of a function $f : \mathcal{X}^n \to \mathbb{R}$ if:*

$$\forall \boldsymbol{z} \in \mathcal{X}^n : LS_f(\boldsymbol{z}) \leq S_{f,t}(\boldsymbol{z});$$

$$\forall \boldsymbol{z}, \boldsymbol{z}' \in \mathcal{X}^n, d(\boldsymbol{z}, \boldsymbol{z}') = 1 : S_{f,t}(\boldsymbol{z}) \leq e^t \cdot S_{f,t}(\boldsymbol{z}').$$

where $d(\boldsymbol{z}, \boldsymbol{z}')$ is the distance between datasets $\boldsymbol{z}$ and $\boldsymbol{z}'$.

Let $\mathcal{Z}_k : \{(x_i, y_i)\}_{i=1}^n \in (\mathbb{R} \times \mathbb{R})^n \to \mathbb{R}^{kn/2}$ denote the function that transforms a set of point coordinates into estimates for each pair of points in our k matchings. The

function that we are concerned with the smooth sensitivity of is med$\circ\mathcal{Z}_k$. We will use the following smooth upper bound to the local sensitivity:

Lemma 10. *Let $z_1 \leq z_2 \leq \cdots \leq z_{2m}$ be a sorting of $\mathcal{Z}_k(\boldsymbol{x}, \boldsymbol{y})$. Then*

$$S^k_{med\circ\mathcal{Z},t}((\boldsymbol{x},\boldsymbol{y})) = \max\Big\{z_{m+k} - z_m, z_m - z_{m-k}, \max_{l=1,\ldots,n}\max_{s=0,\cdots,k(l+1)} e^{-lt}(z_{m+s} - z_{m-(k(l+1)+s)})\Big\},$$

is a t-smooth upper bound on the local sensitivity of med$\circ\mathcal{Z}_k$.

Proof. Proof in the Appendix. □

The algorithm then adds noise proportional to $S^k_{\text{med}\circ\mathcal{Z},t}((\mathbf{x},\mathbf{y}))/\varepsilon$ to med$\circ\mathcal{Z}(\mathbf{x},\mathbf{y})$. Pseudo-code is given in the Appendix.The noise is sampled from the Student's T distribution. There are several other valid choices of noise distributions (see [23] and [9]), but we found the Student's T distribution to be preferable as the mechanism remains stable across values of ε

6.3 DP Gradient Descent

Ordinary Least Squares (OLS) for simple 1-dimensional linear regression is defined as the solution to the optimization problem in Equation 1.There has been an extensive line of work on solving convex optimization problems in a differentially private manner. We use the private gradient descent algorithm of [6, 27] to provide private estimates of the 0.25, 0.75 predictions $(\text{p}_{25}, \text{p}_{75})$. This algorithm performs standard gradient descent, except that noise is added to a clipped version of the gradient at each round (clipped to range $[-\tau, \tau]^2$ for some setting of $\tau > 0$).

6.4 DPIntercept

We also compare the above DP mechanisms to simply adding noise to the average y-value. For any given dataset $(\mathbf{x}, \mathbf{y})$, this method clips the datapoints to a given range $[r_l, r_u]$, computes $\bar{y} = \frac{1}{n}\sum_{i=1}^n y_i$ and outputs a noisy estimate $\tilde{y} = \bar{y} + \text{Lap}\left(0, \frac{r_u - r_l}{\varepsilon n}\right)$ as the predicted $\widetilde{p}_{25}, \widetilde{p}_{75}$ estimates. This method performs well when the slope α is very small.

6.5 A Note on Hyperparameters

We leave the question of how to choose the optimal hyperparameters for each algorithm to future work. Unfortunately, since the optimal hyperparameter settings may reveal sensitive information about the dataset, one *cannot* simply tune the hyperparameters on a subset of the sensitive data. However, we found that for most of the hyperparameters, once a good choice of hyperparameter setting was found, it could be used for a variety of similar datasets. Thus, one realistic way to tune the parameters may be to tune on a public dataset similar to the dataset of interest. For example, for applications using census data, one could tune the parameters on previous years' census data.

7 Experiments and Results

In our experiments, we evaluate the following algorithms: `DPSuffStats`, `DPIntercept`, `DPExpTheilSen`, `DPWideTheilSen`, `DPSSTheilSen`, `DPGradDescent`, and `OpportunityInsights (OI)`. We will present results on a simulated version of the data used by the Opportunity Insights team in creating the Opportunity Insights tool, a real UCI dataset, and synthetically generated datasets that allow us to explore how properties of the data affect relative performance of the algorithms. Our experiments indicate that for a wide range of realistic datasets, and moderate values of ε, it is possible to choose a DP linear regression algorithm where the error due to privacy is less than the standard error. In our motivating use-case of the Opportunity Atlas, we can design a differentially private algorithm that outperforms the heuristic method used by the Opportunity Insights team. This is promising, since the error added by the heuristic method was deemed acceptable for deployment of the Opportunity Atlas, and for use by policy makers.

One particular differentially private algorithm of the robust variety, called `DPExpTheilSen`, emerges as the best algorithm in a wide variety of settings for this small-dataset regime. This algorithm uses the *exponential mechanism* [21] for the differentially private median computation to be used in the Theil-Sen method. We will discuss some reasons for `DPExpTheilSen`'s strong performance, and identify some regimes in which other algorithms perform better.

7.1 Experimental Setup

We will focus on data with values bounded between 0 and 1, so $0 \leq x_i, y_i \leq 1$ for $i = 1, \ldots, n$. This boundedness assumption is inherited from our main use-case, the Opportunity Atlas tool. Not all our methods require this assumption. Note that the independent variables x_i in all the datasets listed below are drawn from different distributions.

7.1.1 Simulated Opportunity Atlas Datasets

The first set of datasets we consider is a simulated version of the data used by the Opportunity Insights team in creating the Opportunity Atlas tool described in Section 3. These datasets are valuable for our evaluation because they mimic a real-world release of privacy-protected statistics. The independent variables x_i of the simulated datasets follow a lognormal distribution. Each datapoint, $(x_i, y_i) \in [0,1]^2$, corresponds to a pair, (parent income percentile rank, child income percentile rank). In the data, every state in the United States is partitioned into small neighborhood-level blocks called tracts. As each individual exists in only one tract, the privacy loss on the entire release is only the maximum privacy loss of the per-tract releases (by the parallel composition theorem of differential privacy [22]). We perform the linear regression on each tract individually. Each of the tracts can look very different, as the distribution of income both across tracts and inside each tract varies in different states. For example, in Illinois, more than 90% of residents have incomes above the national median income. In North Carolina, on the other hand, about 50% of residents have incomes below the national median income. The "best" differentially private algorithm differs from state to state, and even tract to tract. We display results for Illinois (IL) which has a total of $n = 219,594$ datapoints divided among 3,108 tracts. The individual datasets (corresponding to tracts in the Census data) each contain between $n = 30$ and $n = 400$ datapoints. The statistic $n\text{var}(\mathbf{x})$ ranges between 0 and 25, with the majority of tracts having $n\text{var}(\mathbf{x}) \in [0,5]$.

7.1.2 Stock Exchange UCI Dataset

Next, we consider a real dataset that studies the relationship between the Istanbul Stock Exchange and the USD Stock Exchange [2]. It compares $\{x_i =$ Istanbul Stock Exchange national 100 index$\}$ to $\{y_i =$ the USD International Securities Exchange$\}$. The independent variables x_i in this dataset follow a normal distribution. This dataset has $n = 250$ datapoints.

7.1.3 Other Real Datasets

In the full version of this paper, we evaluate the DP algorithms on two additional real datasets – a Washington DC Bikeshare UCI dataset and a Carbon Nanotubes UCI dataset. The Washington DC Bikeshare UCI dataset is a family of 288 small datasets (with sizes ranging from 45 to 62 datapoints) that contain the temperature (x_i) and user count of a bikeshare program (y_i) for a fixed time period. The independent variables x_i have a normal distribution. The Carbon Nanotubes dataset is a larger dataset (with $10,683$ datapoints), where x_i is the u-coordinate of the initial atomic coordinates of a molecule and y_i is the u-coordinate of the calculated atomic coordinates after the energy of the system has been minimized. The independent variables x_i have a uniform distribution. Due to space constraints and for clarity of exposition, we only briefly mention the results on these additional real datasets.

7.1.4 Synthetic Datasets

Finally, we construct synthetic datasets by sampling $x_i \in \mathbb{R}$, for $i = 1, \ldots, n$, independently from a uniform distribution with $\bar{x} = 0.5$ and variance σ_x^2. For each x_i, the corresponding y_i is generated as $y_i = \alpha x_i + \beta + e_i$, where $\alpha = 0.5, \beta = 0.2$, and e_i is sampled from $\mathcal{N}(0, \sigma_e^2)$. The (x_i, y_i) datapoints are then clipped to the box $[0,1]^2$. The DP algorithms estimate the prediction at x_{new} using privacy parameter ε.

The values of n, σ_x^2, σ_e^2, x_{new}, and ε vary across the individual experiments. The synthetic data experiments are designed to study which properties of the data and privacy regime determine the performance of the private algorithms. Thus, in these experiments, we vary one of parameters listed above and observe the impact on the accuracy of the algorithms and their relative performance to each other. Since we know the ground truth on this synthetically generated data, we plot empirical error bounds that take into account both the sampling error and the error due to the DP algorithms, $\hat{C}_{\text{true}}(0.68)/\sigma(\widehat{p}_{x_{new}})$. We evaluate `DPTheilSenkMatch` with $k = 10$ in the synthetic experiments rather than `DPTheilSen`, since the former is computationally more

efficient and still gives us insight into the performance of the latter.

7.1.5 Privacy Parameters and Hyperparameters

A note on the privacy parameters in the experiments: We will state the privacy budget used to compute the pair (p_{25}, p_{75}), but we will only show empirical error bounds for p_{25}. The empirical error bounds for p_{75} display similar phenomena. The algorithms `DPSuffStats` and `DPGradDescent` inherently release both point estimates together so the privacy loss is the same whether we release the pair (p_{25}, p_{75}) or just p_{25}. However, `DPExpTheilSen`, `DPWideTheilSen`, `DPSSTheilSen` and the `OI` algorithm use half their budget to release p_{25} and half their budget to release p_{75}, separately.

Our experiments explore a range of privacy parameters. The synthetic data experiments show results using moderate to small privacy loss parameters (Figures 5a, 5b, and 6 use $\varepsilon = 1$, and Figure 5c tests ε values ranging from 0.01 to 10). The evaluation on the Stock Exchange UCI dataset uses a moderate privacy loss parameter of $\varepsilon = 2$. For the simulated Opportunity Atlas data, we used a large privacy loss parameter $\varepsilon = 8$ ($\varepsilon = 16$ for both $\widetilde{p}_{25}$ and $\widetilde{p}_{75}$) in order to match what was used in the real-world deployment.

Hyperparameters were tuned on the semi-synthetic Opportunity Insights data by experimenting with many different choices, and choosing the best. The hyperparameters are listed in Table 1. We leave the question of optimizing hyperparameters in a privacy-preserving way to future work. Hyperparameters for the synthetic datasets were chosen according to choices that seemed to perform well on the Opportunity Insights datasets. We empirically observe that good parameter choices tended to be good for a wide variety of datasets. (In particular, our synthetic data and our two real dataset examples all have different x distributions. This ensures that the hyperparameters were not tuned to the specific datasets, but also leaves some room for improvement in the performance by more careful setting of hyperparameters.) As mentioned above, one realistic way to tune the parameters may be to tune on a public dataset similar to the dataset of interest.

Table 1. Hyperparameters used in experiments on OI data, UCI datasets, and synthetic data.

Algorithms	OI data	Synthetic data
`DPSuffStats`	$r_l = 0, r_u = 1$	$r_l = 0, r_u = 1$
`DPIntercept`	$r_l = 0, r_u = 1$	N/A
`DPExpTheilSen`	$r_l = -0.5, r_u = 1.5$	$r_l = -2, r_u = 2$
`DPWideTheilSen`	$r_l = -0.5, r_u = 1.5$, $\theta = 0.01$	$r_l = -2, r_u = 2$, $\theta = 0.01$
`DPSSTheilSen`	$r_l = -0.5, r_u = 1.5$, $d = 3$	$r_l = -2, r_u = 2$, $d = 3$
`DPGradDescent`	$\tau = 1$, T = 80	$\tau = 1$, T = 80

7.2 Results on Simulated and Real Datasets

7.2.1 Simulated Opportunity Atlas Datasets

Figure 3 shows the results of all the DP linear regression algorithms, as well as the heuristic mechanism used by the Opportunity Insights team (labeled `OI`), on the Opportunity Insights simulated data for the state of Illinois For each algorithm, we build a cumulative distribution function of the empirical error bounds set over the tracts in that state. The vertical dotted line in Figures 3 intercepts each curve at the point where the *noise due to privacy exceeds the standard error*.

The privacy-loss parameter of $\varepsilon = 16$ used in the Atlas was selected by the Opportunity Insights team and a Census Disclosure Review Board by balancing the privacy and utility considerations. Although we use the same value in our experiments for sake of comparison, we stress that we generally do not recommend or endorse using such a large privacy-loss parameter in applications.

Figure 3 shows that there exist differentially private algorithms that are competitive with, and in many cases more accurate than, the algorithm currently de-

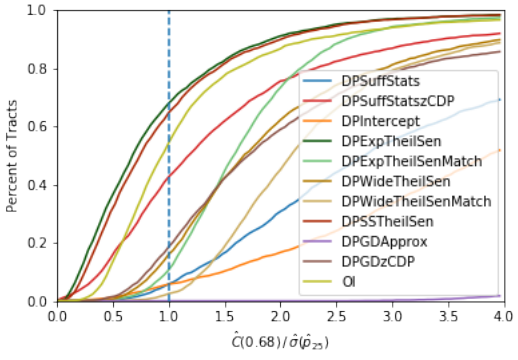


Fig. 3. Empirical CDF for the Empirical 0.68 error bounds, $\hat{C}(0.68)$, normalized by empirical OLS standard error when evaluated on Opportunity Insights data for the state of Illinois. Privacy parameter $\varepsilon = 16$ for the pair $(\widetilde{p}_{25}, \widetilde{p}_{75})$.

ployed in the Opportunity Atlas, such as DPExpTheilSen and DPSSTheilSen. Additionally, we note that the methods used by Opportunity Insights are highly tailored to their setting; as discussed in Section 4, the computation of an upper bound on the local sensitivity relies on coordination across Census tracts. (This is why we are not able to include this algorithm in our stock exchange data or synthetic data experiments – there is no natural analogue for this algorithm in the context of a single dataset.) The differentially private methods, on the other hand, are general-purpose and do not require any coordination across tracts.

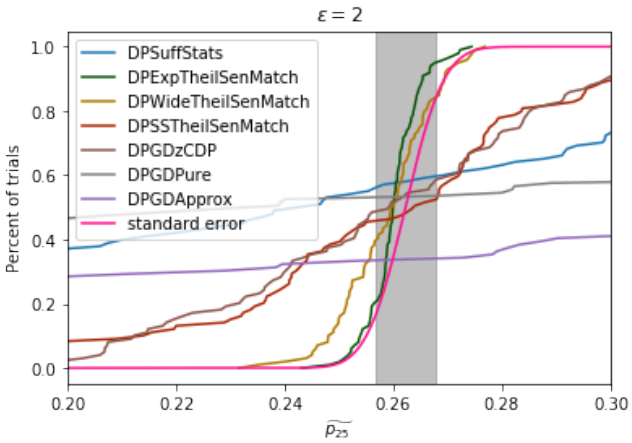


Fig. 4. Stock Exchange UCI Data. Empirical cdf of the output distribution of the estimate of p_{25} after 100 trials of each algorithm with $\varepsilon = 2$. The grey region includes all the values that are within one standard error of $\widehat{\sigma}(\widehat{p}_{25})$. The curve labelled "standard error" shows the non-private posterior belief on the value of the p_{25} assuming Gaussian noise.

7.2.2 Results on Stock Exchange UCI Dataset

Next, we provide experimental results of all the DP linear regression algorithms on the Stock Exchange UCI dataset. We use the same hyperparameter settings for these datasets that were used for the OI dataset, shown in Table 1. Figure 4 shows the empirical cumulative density function of the output distribution on the Stock Exchange dataset. This has a different form from that of the empirical CDF which appears in Figure 3. On this real dataset, for a moderate ε value of 2, the additional noise due to privacy using DPExpTheilSen is less than the standard error. Similar to Figure 3, Figure 4 shows that DPExpTheilSen is generally the best performing algorithm, followed by DPWideTheilSen and DPSSTheilSen.

7.2.3 Results on Additional Real Datasets

The results on the Washington DC Bikeshare dataset and the Carbon Nanotubes dataset corroborate our findings on the Opportunity Atlas and Stock Exchange datasets. For a range of realistic ε values, the additional noise due to privacy, in particular when using DPExpTheilSen, remains less than the standard error. The full version of our paper contains detailed analysis of the results on these additional real datasets, but we will not include any further discussion here due to space constraints and for clarity of exposition.

7.3 Robustness vs. Non-robustness: Guidance for Algorithm Selection

The DP algorithms we evaluate can be divided into two classes, *robust* DP estimators based on Theil-Sen — DPSSTheilSen, DPExpTheilSen and DPWideTheilSen — and *non-robust* DP estimators based on OLS — DPSuffStats and DPGradDescent. Experimentally, we found that the algorithms' behaviour tends to be clustered in these two classes, with the robust estimators outperforming the non-robust estimators in a wide variety of parameter regimes. In the experiment we saw in the previous section (Figure 3), DPExpTheilSen was the best performing algorithm, followed by DPWideTheilSen and DPSSTheilSen. In our experiments on synthetic data, however, we will see that the non-robust estimators outperform the robust estimators in some parameter regimes.

7.3.1 DPSuffStats and DPExpTheilSen

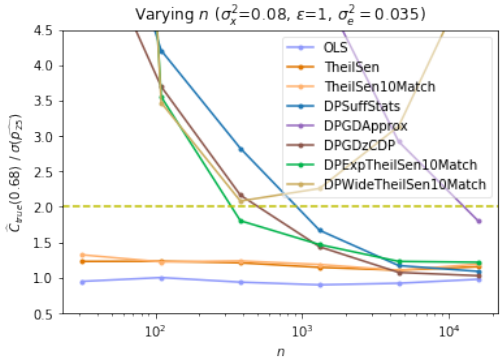
In Figure 5, we investigate the relative performance $(\hat{C}_{\text{true}}(0.68)/\sigma(\widehat{p}_{25}))$ of the algorithms in several parameter regimes of n, ε, and σ_x^2 on synthetically generated data. For each parameter setting and for each algorithm, we plot of the average value of $\hat{C}_{\text{true}}(0.68)/\sigma(\widehat{p}_{25})$ over 50 trials on a single dataset, and average again over 500 independently sampled datasets. Across ranges for each of these three parameters ($n \in [31, 15848]$;

$\sigma_x^2 \in [0.003, 0.08]$; $\varepsilon \in [0.01, 10]$, all varied on a logarithmic scale), we see that DPExpTheilSen10Match[5] or DPSuffStats is consistently close to the best performing algorithm. We see that DPGDzCDP and DPSuffStats trend towards taking over as the best algorithm as ε, n, and σ_x^2 increases.

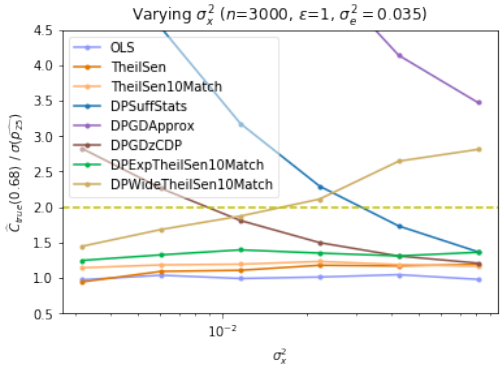
For parameter regimes in which the non-robust algorithms outperform the robust estimators, DPSuffStats is preferable to DPGDzCDP since it is more efficient, requires less hyperparameter tuning (except for the clipping bound $[r_l, r_u]$ on the inputs x_i and y_i for all $i \in [n]$), satisfies a stronger privacy guarantee ("pure DP" instead of "concentrated DP"), and releases the noisy sufficient statistics with no additional cost in privacy. Experimentally, we find that the main indicator for deciding between robust estimators and non-robust estimators is the quantity $\varepsilon n \text{var}(\mathbf{x})$ (which is a proxy for $\varepsilon n \sigma_x^2$). Roughly, when $\varepsilon n \text{var}(\mathbf{x})$ and σ_e^2 are both large, we conclude that DPSuffStats is the best choice among the DP algorithms tested; otherwise, the robust estimator DPExpTheilSen10Match typically has lower error. Hyperparameter tuning and the quantity $|x_{new} - \bar{x}|$ also play minor roles in determining the relative performance of the algorithms.

7.3.2 Slopes vs. Point Estimates

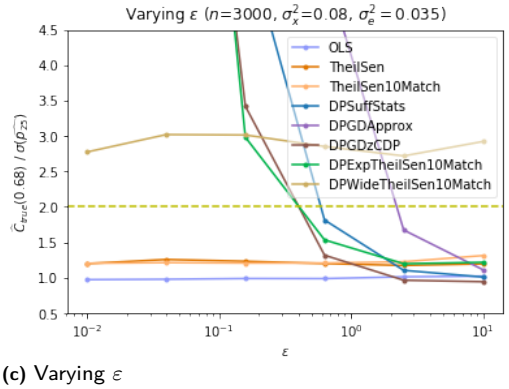
In Figure 7, we show error bounds with respect to the slopes instead of point estimates as we vary ε. The slopes here are computed directly (e.g., for the Theil-Sen algorithms, we compute the DP median of many estimates of the slope); we could also compute the slopes using the p_{25} and p_{75} estimates. We observe similar relative performance of the algorithms between the slopes and point estimates (i.e., between Figure 7 and Figure 5c). We note that for the Theil-Sen based algorithms, it can be more difficult to choose hyperparameters for the slopes as these can be unbounded while the point estimates are bounded between 0 and 1 in our setting of data normalized to $[0, 1]$. As we show in the full version of the paper, however, the Theil-Sen algorithms are less sensitive to the choice of hyperparameters than DPSuffStats.



(a) Varying n



(b) Varying σ_x^2



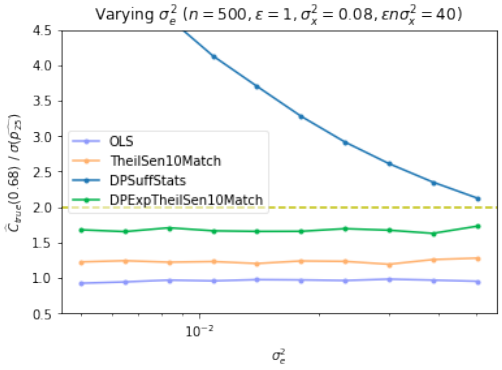
(c) Varying ε

Fig. 5. Relative error ($\widehat{C}_{\text{true}}/\sigma(\widehat{p}_{25})$) of DP and non-private algorithms on synthetic data as n varies from 31 to 15,848, σ_x^2 from 0.003 to 0.08, and ε from 0.01 to 10.

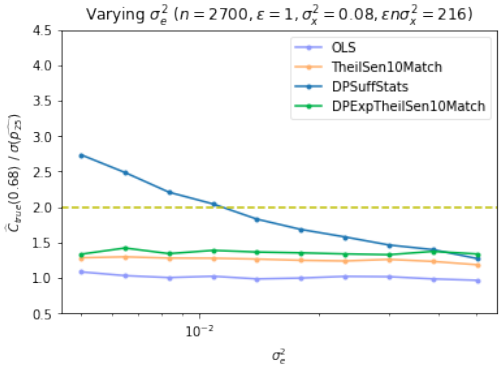
7.3.3 Role of εnvar(x)

In the experiments of Figure 6, we control the quantity $\varepsilon n \sigma_x^2$, which combines the three parameters varied separately in Figure 5 — the size of the dataset, how concentrated the independent variable of the data is and how private the mechanism is, and is a "ground truth" analogue of the empirical quantity εnvar$\mathbf{x}$ that appears in OLS. It appears to be a better indicator of the performance of DP mechanisms than any of the individual statistics ε, n, or σ_x^2 in isolation. In Fig-

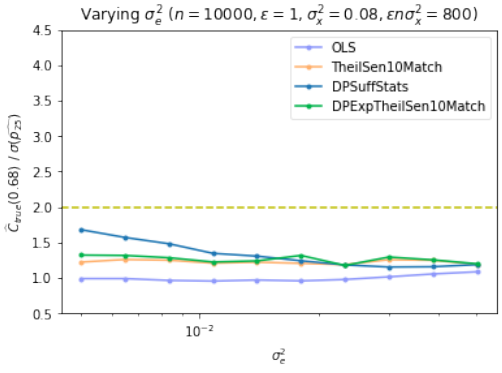
5 DPExpTheilSen10Match is DPExpTheilSen where instead of considering all pairs of points, we use 10 random matchings of the points. We use it in the synthetic experiments since it is more computationally efficient than, but similarly performing to, DPTheilSen.



(a) $\varepsilon n\sigma_x^2 = 40$; varying σ_e^2



(b) $\varepsilon n\sigma_x^2 = 216$; varying σ_e^2



(c) $\varepsilon n\sigma_x^2 = 800$; varying σ_e^2

Fig. 6. Comparing `DPSuffStats` and `DPExpTheilSen10Match` on synthetic data as σ_e^2 varies from 0.005 to 0.05. Plotting $\hat{C}_{\text{true}}/\sigma(\widehat{p_{25}})$. OLS and TheilSen10Match included for reference.

ure 6, we compare the performance of `DPSuffStats` and `DPExpTheilSen10Match` when we hold $\varepsilon n\sigma_x^2$ constant and vary σ_e^2, the variance of the noise e_i in the linear relationship $y_i = \alpha x_i + \beta + e_i$. The error is computed as the average error over 20 trials and 500 independently sampled datasets. Notice that the periwinkle line presents the error of the non-private OLS estimator, which is our baseline. In all of our synthetic data experiments, in which the e_i's are Gaussian, once $\varepsilon n\sigma_x^2 > 400$ and $\sigma_e^2 \geq 0.03$, `DPSuffStats` is the better performing algorithm. It is also important to note that once $\varepsilon n\text{var}(\mathbf{x})$

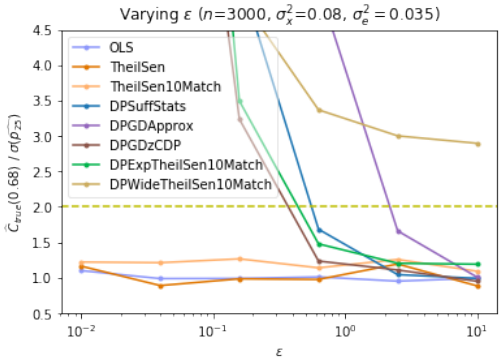


Fig. 7. Relative error of DP and non-private algorithms as ε varies. In this figure, the error is measured with respect to the slopes not point estimates.

is large, both `DPSuffStats` and `DPExpTheilSen` perform well.

The error, as measured by $\hat{C}_{\text{true}}(0.68)$, of both non-private OLS and Theil-Sen estimators converges to 0 as $n \to \infty$ at the same asymptotic rate. However, OLS converges a constant factor faster than Theil-Sen, which can be seen by the fact that the Theil-Sen and TheilSen10Match lines are strictly above the OLS line in Figures 5 and 6. As $\varepsilon n\sigma_x^2$ increases, `DPSuffStats` and `DPGD` approach the performance of non-private OLS, and hence outperform both the non-private and private versions of Theil-Sen.

Additional experiments we ran, which we were not able to include here to due to space constraints, confirm that $\varepsilon n\sigma_x^2$ is a strong indicator of the relative performance of `DPSuffStats` and `DPExpTheilSen` as long as σ_e is not too small, even as other variables in the OLS standard error equation (3) – including the difference between x_{new} and the mean of the x values, $|x_{new} - \bar{x}|$ – are varied.

7.3.4 The Role of Hyperparameter Tuning

A final major distinguishing feature among the `DPTheilSen` algorithms, `DPSuffStats` and `DPGradDescent` is the amount of prior knowledge needed by the data analyst to choose the hyperparameters appropriately. Notably, `DPSuffStats` does not require any hyperparameter tuning other than a bound on the data. The `DPTheilSen` algorithms require some reasonable knowledge of the range that p_{25} and p_{75} lie in. `DPGradDescent` requires some knowledge of where the input values lie so it can set its hyperparameters τ, T.

7.3.5 Which Robust Estimator?

In the majority of the regimes we have tested, `DPExpTheilSen` outperforms all the other private algorithms. However, another variation on DP TheilSen, `DPWideTheilSen`, can outperform `DPExpTheilSen` when the standard error is small. When there is little noise in the data we expect the set of pairwise estimates to be highly concentrated. As discussed in Section 6.2, this is a difficult setting for `DPExpTheilSen`; we designed `DPWideTheilSen` to address this problem.

8 Conclusion

It is possible to design DP simple linear regression algorithms where the distortion added by the private algorithm is less than the standard error, even for small datasets. In this work, we found that in order to achieve this we needed to switch from OLS regression to the more robust linear regression estimator, Theil-Sen. We identified key factors that analysts should consider when deciding whether DP methods based on robust or non-robust estimators are right for their applications.

This work is the first to experimentally evaluate the performance of robust algorithms for differentially private linear regression. Prior theoretical work has highlighted the strong connection between robust statistics and differential privacy [15, 18], but experimental evaluations of differentially private algorithms for linear regression (such as the systematic study by Wang [29]) have failed to include robust algorithms in their comparisons. In addition, we are the first to consider the winning algorithm from our experiments, `DPTheilSen`, under privacy constraints. This algorithm is a generalization of the "Short-Cut Regression Algorithm" that was analyzed theoretically (but not experimentally) by Dwork and Lei [18].

The focus of this work has been on DP univariate regression releases for small datasets. The challenge of selecting an algorithm with good utility for this setting remains a barrier to adoption of differential privacy for many practical applications. Prior work on DP linear regression has focused on multiple linear regression (i.e., where there are several or many independent variables), albeit in an asymptotic or large-dataset setting. Our results show that even in the setting of simple linear regression, the story is already quite nuanced. The algorithm that performs best depends on properties of the dataset, such as $n\text{var}(\mathbf{x})$, which cannot be directly used without violating differential privacy. One has to make the choice based on guesses (e.g., using similar public datasets) or develop differentially private methods for selecting the algorithm, a problem which we leave to future work. However, our experimental evaluation offers valuable heuristics for choosing a suitable algorithm in practice – in particular, our experiments demonstrate that `DPTheilSen` performs well across many regimes (including with small dataset sizes, when the data range is unknown and when knowledge of the output range is limited) compared to `DPSuffStats`, and is therefore a reliable choice for most applications. Our findings highlight important directions for future work, such as generalizing `DPTheilSen` to cover multivariate regression, providing uncertainty estimates such as confidence intervals, and developing theoretical explanations for its superior performance in the small dataset regime.

9 Acknowledgements

AS and AM were supported by NSF awards CCF-1763786 and IIS-1447700, a research award from the Sloan Foundation, and (for AM) the Hariri Institute for Computing. At Northeastern, AM was supported by a Fellowship from the Cybersecurity & Privacy Institute and NSF grant CCF-1750640. SV was supported by a Simons Investigator Award. DA was partially supported by a Fellowship from Facebook. SV, DA, JS, AM and AS were supported by Cooperative Agreement CB16ADR0160001 with the Census Bureau. The views expressed in this paper are those of the authors and not those of the U.S. Census Bureau or any other sponsor.

References

[1] Jacob Abernethy, Chansoo Lee, and Ambuj Tewari. 2016. Perturbation techniques in online learning and optimization. *Perturbations, Optimization, and Statistics* (2016), 233.

[2] Oguz Akbiligic, Hamparsum Bozdogan, and M. Erdal Balaban. 2013. A novel Hybrid RBF Neural Networks model as a forecaster. *Statistics and Computing* (2013). This dataset was collected from imkb.gov.tr and finance.yahoo.com.

[3] Hilal Asi and John C Duchi. 2020. Instance-optimality in differential privacy via approximate inverse sensitivity mechanisms. *Advances in Neural Information Processing Systems* 33 (2020).

[4] Jordan Awan and Aleksandra Slavković. 2020. Structure and sensitivity in differential privacy: Comparing k-norm mechanisms. *J. Amer. Statist. Assoc.* just-accepted (2020), 1–56.

[5] Emily Badger and Quoctrung Bui. 2020. Detailed Maps Show How Neighborhoods Shape Children for Life. https://www.nytimes.com/2018/10/01/upshot/maps-neighborhoods-shape-child-poverty.html. Online; accessed 15 October 2020.

[6] Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. 2014. Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*. 464–473.

[7] Garrett Bernstein and Daniel R Sheldon. 2019. Differentially Private Bayesian Linear Regression. In *Advances in Neural Information Processing Systems 32*. 523–533.

[8] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*. Springer, 635–658.

[9] Mark Bun and Thomas Steinke. 2019. Average-Case Averages: Private Algorithms for Smooth Sensitivity and Mean Estimation. In *Advances in Neural Information Processing Systems 32*. 181–191.

[10] Tony Cai, Yichen Wang, and Linjun Zhang. 2019. The Cost of Privacy: Optimal Rates of Convergence for Parameter Estimation with Differential Privacy. *CoRR* abs/1902.04495 (2019). http://arxiv.org/abs/1902.04495

[11] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. 2011. Differentially Private Empirical Risk Minimization. *Journal of Machine Learning Research* 12 (2011), 1069–1109.

[12] Raj Chetty, John N Friedman, Nathaniel Hendren, Maggie R Jones, and Sonya R Porter. 2018. *The opportunity atlas: Mapping the childhood roots of social mobility*. Technical Report. National Bureau of Economic Research.

[13] Raj Chetty, Nathaniel Hendren, Patrick Kline, and Emmanuel Saez. 2014. Where is the land of opportunity? The geography of intergenerational mobility in the United States. *The Quarterly Journal of Economics* 129, 4 (2014), 1553–1623.

[14] Graham Cormode. [n. d.]. Building Blocks of Privacy: Differentially Private Mechanisms. ([n. d.]), 18–19.

[15] Simon Couch, Zeki Kazan, Kaiyan Shi, Andrew Bray, and Adam Groce. 2019. Differentially Private Nonparametric Hypothesis Testing. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*. 737–751.

[16] Alfred DeMaris. 2004. *Regression with social data : modeling continuous and limited response variables*. Wiley-Interscience, Hoboken, NJ.

[17] Irit Dinur and Kobbi Nissim. 2003. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 202–210.

[18] Cynthia Dwork and Jing Lei. 2009. Differential privacy and robust statistics.. In *STOC*, Vol. 9. 371–380.

[19] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*. 265–284.

[20] Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. 2014. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *STOC*. 11–20.

[21] Frank McSherry and Kunal Talwar. 2007. Mechanism Design via Differential Privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*. 94–103.

[22] Frank D McSherry. 2009. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. 19–30.

[23] Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*. 75–84.

[24] Pranab Kumar Sen. 1968. Estimates of the regression coefficient based on Kendall's tau. *Journal of the American statistical association* 63, 324 (1968), 1379–1389.

[25] Or Sheffet. 2017. Differentially Private Ordinary Least Squares. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 3105–3114. http://proceedings.mlr.press/v70/sheffet17a.html

[26] Adam Smith. 2011. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*. 813–822.

[27] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. 2013. Stochastic gradient descent with differentially private updates. In *IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013, Austin, TX, USA, December 3-5, 2013*. 245–248.

[28] Henri Theil. 1950. A rank-invariant method of linear and polynomial regression analysis, 3; confidence regions for the parameters of polynomial regression equations. *Indagationes Mathematicae* 1, 2 (1950), 467–482.

[29] Yu-Xiang Wang. 2018. 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, UAI 2018, Monterey, California, USA, August 6-10, 2018*. 93–103.

[30] Xin Yan and Xiao Gang Su. 2009. *Linear Regression Analysis: Theory and Computing*. World Scientific Publishing Co., Inc., USA.

[31] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. 2012. Functional Mechanism: Regression Analysis under Differential Privacy. *Proc. VLDB Endow.* 5, 11 (2012), 1364–1375.

A Some Results in Differential Privacy

In this section we will briefly review some of the fundamental definitions and results pertaining to general differentially private algorithms.

For any query function $f : \mathcal{X}^n \to \mathbb{R}^K$ let $\mathrm{GS}_f = \max_{d \sim d'} \|f(d) - f(d')\|$, called the global sensitivity, be the maximum amount the query can differ on neighboring datasets.

Theorem 11 (Laplace Mechanism [19]). *For any privacy parameter $\varepsilon > 0$ and any given query function $f : \mathcal{X}^n \to \mathbb{R}^K$ and database $d \in \mathcal{X}^n$, the* Laplace mechanism *outputs $\tilde{f}_L(d) = f(d) + (R_1, \ldots, R_K)$, where $R_1, \ldots, R_K \sim Lap(0, \frac{GS_f}{\varepsilon})$ are i.i.d. random variables drawn from the 0-mean Laplace distribution with scale $\frac{GS_f}{\varepsilon}$. The Laplace mechanism is $(\varepsilon, 0)$-DP.*

Theorem 12 (Exponential Mechanism [21]). *Given an arbitrary range $\mathcal{R}$, let $u : \mathcal{X}^n \times \mathcal{R} \to \mathbb{R}$ be a utility function that maps database/output pairs to utility scores. Let $GS_u = \max_r GS_{u(\cdot,r)}$. For a fixed database $d \in \mathcal{X}^n$ and privacy parameter $\varepsilon > 0$, the* exponential mechanism *outputs an element $r \in \mathcal{R}$ with probability proportional to $\exp\left(\frac{\varepsilon \cdot u(d,r)}{2GS_u}\right)$. The exponential mechanism is $(\varepsilon, 0)$-DP.*

The following results allow us to use differentially private algorithms as building blocks in larger algorithms.

Lemma 13 (Post-Processing [19]). *Let $M : \mathcal{X}^n \to \mathcal{Y}$ be an (ε, δ) differentially private and $f : \mathcal{Y} \to \mathcal{R}$ be a (randomized) function. Then $f \circ M : \mathcal{X}^n \to \mathcal{R}$ is an (ε, δ) differentially private algorithm.*

Theorem 14 (Basic Composition [19]). *For any $k \in [K]$, let M_k be an $(\varepsilon_k, \delta_k)$ differentially private algorithm. Then the composition of the T mechanisms $M = (M_1, \ldots, M_K)$ is (ε, δ) differentially private where $\varepsilon = \sum_{k \in [K]} \varepsilon_k$ and $\delta = \sum_{k \in [K]} \delta_k$.*

Definition 15 (Coupling). *Let $\boldsymbol{z}$ and $\boldsymbol{z}'$ be two random variables defined over the probability spaces Z and Z', respectively. A coupling of $\boldsymbol{z}$ and $\boldsymbol{z}'$ is a joint variable $(\boldsymbol{z}_c, \boldsymbol{z}'_c)$ taking values in the product space $(Z \times Z')$ such that $\boldsymbol{z}_c$ has the same marginal distribution as $\boldsymbol{z}$ and $\boldsymbol{z}'_c$ has the same marginal distribution as $\boldsymbol{z}'$.*

Definition 16 (c-Lipschitz randomized transformations). *A randomized transformation $T : \mathcal{X}^n \to \mathcal{Y}^m$ is c-Lipschitz if for all datasets $d, d' \in \mathcal{X}^n$, there exists a coupling $(\boldsymbol{z}_c, \boldsymbol{z}'_c)$ of the random variables $\boldsymbol{z} = T(d)$ and $\boldsymbol{z}' = T(d')$ such that with probability 1, $H(\boldsymbol{z}_c, \boldsymbol{z}'_c) \leq c \cdot H(d, d')$ where H denotes Hamming distance.*

Lemma 17. (Composition with Lipschitz transformations (well-known)) *Let M be an (ε, δ)-DP algorithm, and let T be a c-Lipschitz transformation of the data with respect to the Hamming distance. Then, $M \circ T$ is $(c\varepsilon, \delta)$-DP.*

Proof. The lemma follows directly from the Lipschitz property on adjacent databases and the definition of (ε, δ)-differential privacy. □

B Privacy Proof of `DPSuffStats`

Lemma 18. *Suppose we are given dataset $\boldsymbol{x}, \boldsymbol{y} \in [r_l, r_u]^n$ where $0 \leq r_l \leq r_u$. Let*

$$ncov(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x} - \bar{x}\mathbf{1}, \boldsymbol{y} - \bar{y}\mathbf{1} \rangle = (\sum_{i=1}^{n} x_i \cdot y_i) - \frac{(\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n},$$

and

$$nvar(\boldsymbol{x}) = \langle \boldsymbol{x} - \bar{x}\mathbf{1}, \boldsymbol{x} - \bar{x}\mathbf{1} \rangle = (\sum_{i=1}^{n} x_i^2) - \frac{(\sum_{i=1}^{n} x_i)^2}{n}.$$

Also, let $\bar{x}, \bar{y}$ be the means of $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively and $\mathbf{1}$ be the all ones vector.

Then if GS_{ncov} and GS_{nvar} are the global sensitivities of functions ncov and nvar then $GS_{ncov} = \left(1 - \frac{1}{n}\right) r_u^2$ and $GS_{nvar} = \left(1 - \frac{1}{n}\right) r_u^2$.

Proof. Let $\mathbf{z} = \langle \mathbf{x}, \mathbf{y} \rangle$ and $\mathbf{z}' = \langle \mathbf{x}', \mathbf{y}' \rangle$ be neighbouring databases differing on the nth datapoint [6]. Let $a = \sum_{i=1}^{n-1} x_i$ and $b = \sum_{i=1}^{n-1} y_i$ and note that $\max\{a, b\} \leq (n-1)$. Then,

$$\begin{aligned} &n\mathrm{var}(\mathbf{x}) - n\mathrm{var}(\mathbf{x}') \\ &= x_n^2 - x_n'^2 - \frac{2ax_n}{n} - \frac{x_n^2}{n} + \frac{2ax_n'}{n} + \frac{x_n'^2}{n} \\ &= (1 - \frac{1}{n})(x_n^2 - x_n'^2) + \frac{2a}{n}(x_n' - x_n). \end{aligned}$$

If $x_n' - x_n \leq 0$ then $n\mathrm{var}(\mathbf{x}) - n\mathrm{var}(\mathbf{x}') \leq (1 - \frac{1}{n})(x_n^2 - x_n'^2) \leq (1 - \frac{1}{n}) r_u^2$. Otherwise,

$$\begin{aligned} &n\mathrm{var}(\mathbf{x}) - n\mathrm{var}(\mathbf{x}') \\ &\leq (1 - \frac{1}{n})(x_n^2 - x_n'^2) + \frac{2(n-1)}{n}(x_n' - x_n) \\ &= (1 - \frac{1}{n})(x_n^2 - 2x_n + 2x_n' - x_n'^2) \\ &= (1 - \frac{1}{n})((x_n^2 - 2x_n) - (x_n'^2 - 2x_n')) \end{aligned}$$

6 This is without loss of generality as we can always "rotate" both databases until the index on which they differ becomes the nth datapoint.

Since $x_n \in [r_l, r_u]$ we have $x_n^2 - 2x_n \leq \max(r_l(r_l - 2), r_u(r_u - 2))$, so $n\text{var}(\mathbf{x}) - n\text{var}(\mathbf{x}') \leq (1 - \frac{1}{n})r_u^2$.

Also,

$$\begin{aligned} &n\text{cov}(\mathbf{x}, \mathbf{y}) - n\text{cov}(\mathbf{x}', \mathbf{y}') \\ &= x_n y_n - x_n' y_n' + \\ &\frac{a(y_n' - y_n) + b(x_n' - x_n) + x_n' y_n' - x_n y_n}{n} \\ &\leq (1 - \frac{1}{n})(x_n y_n - x_n' y_n') + \\ &\frac{a(y_n' - y_n) + b(x_n' - x_n)}{n} \end{aligned}$$

If $y_n' - y_n \leq 0$ and $x_n' - x_n \leq 0$ then $n\text{cov}(\mathbf{x}, \mathbf{y}) - n\text{cov}(\mathbf{x}', \mathbf{y}') \leq (1 - \frac{1}{n})(x_n y_n - x_n' y_n') \leq (1 - \frac{1}{n})r_u^2$. If $y_n' - y_n \leq 0$ and $x_n' - x_n > 0$ then $n\text{cov}(\mathbf{x}, \mathbf{y}) - n\text{cov}(\mathbf{x}', \mathbf{y}') \leq (1 - \frac{1}{n})(x_n y_n - x_n' y_n' + (x_n' - x_n)) \leq (1 - \frac{1}{n})r_u^2$. For similar sub-cases, we obtain that $n\text{cov}(\mathbf{x}, \mathbf{y}) - n\text{cov}(\mathbf{x}', \mathbf{y}') \leq (1 - \frac{1}{n})r_u^2$. □

Proof of Lemma 4: (`DPSuffStats`*).* By Lemma 18, we have that the global sensitivity of both $n\text{cov}(\mathbf{x}, \mathbf{y})$ and $n\text{var}(\mathbf{x})$ is bounded by $\Delta = (1 - 1/n)\, r_u^2$. Therefore, if we sample $L_1, L_2 \sim \text{Lap}(0, 3\Delta/\varepsilon)$ then both $n\text{cov}(\mathbf{x}, \mathbf{y}) + L_1$ and $n\text{var}(\mathbf{x}) + L_2$ are $(\varepsilon/3, 0)$-DP estimates by Theorem 11. By the post-processing properties of differential privacy (Lemma 13), $1/(n\text{var}(\mathbf{x}) + L_2)$ is a private release and the test $n\text{var}(\mathbf{x}) + L_2 > 0$ is also private. As a result, $\tilde{\alpha}$ is a $(2\varepsilon/3, 0)$-DP release. Now to calculate the private intercept $\tilde{\beta}$, we use the global sensitivity of $(\bar{y} - \tilde{\alpha}\bar{x})$ which is at most $r_u/n \cdot (1 + |\tilde{\alpha}|)$, since the means of $\mathbf{x}, \mathbf{y}$ can change by at most r_u/n. [7] The Laplace noise we add ensures the private release of the intercept is $(\varepsilon/3, 0)$-DP. By composition properties of differential privacy (Theorem 14), Algorithm 1 is $(\varepsilon, 0)$-DP. □

C `DPExpTheilSen` and `DPWideTheilSen`

C.1 Privacy Proofs for `DPExpTheilSen` and `DPWideTheilSen`

Lemma 19. *Let T be the following randomized algorithm. For dataset $d = (x_i, y_i)_{i=1}^n$, let $K_n(d)$ be the complete graph on the n datapoints, where edges denote points paired together to compute estimates in Theil-Sen. Then, from $K_n(d)$ we can randomly select k maximal matchings, $\tau_1, \ldots, \tau_{n-1}$, where each τ_i is a vector of $\lfloor n/2 \rfloor$ vectors of size 2. Suppose $T(d)$ uses these k matchings to compute the corresponding pairwise estimates (up to $k\lfloor n/2 \rfloor$ estimates). Then T is a k-Lipschitz randomized transformation.*

Proof. Let $\mathbf{z} = T(d)$ and $\mathbf{z}' = T(d')$ denote the multi-sets of estimates that result from applying T to datasets d and d', respectively. We can define a coupling $\mathbf{z}_c$ and $\mathbf{z}_c'$ of $\mathbf{z}$ and $\mathbf{z}'$. First, use k matchings, $\tau_1, \ldots, \tau_k$, to compute the multi-set of estimates $\mathbf{z}_c = \{z_{j,l}^{(p_{xnew})} : (x_j, x_l) \in \Sigma_1 \cup \ldots \cup \Sigma_k\}\}$. Now, using the same method of selection, choose the corresponding k matchings from $K_n(d')$ to compute a multi-set of estimates $\mathbf{z}_c' = \{z_{j,l}^{(p_{xnew})} : (x_j', x_l') \in \Sigma_1 \cup \ldots \cup \Sigma_k\}$. This is a valid coupling because the k matchings are randomly sampled using the same process from the complete graphs $K_n(d)$ and $K_n(d')$, respectively, matching the marginal distributions of $\mathbf{z}$ and $\mathbf{z}'$. Notice that every datapoint x_j is used to compute exactly k estimates in $\mathbf{z}_c$. Therefore, for every datapoint at which d and d' differ, $\mathbf{z}_c$ and $\mathbf{z}_c'$ differ by at most k estimates. Therefore, by the triangle inequality, we are done. □

Proof of Lemma 5. If $\text{DPmed}(z^{(p25)}, \varepsilon, (n, k, \text{hyperparameters})) = \mathcal{M}(z^{(p25)}, \text{hyperparameters}))$ then Algorithm 2 is a composition of two algorithms, $\mathcal{M} \circ T$, where by Lemma 19, T is a k-Lipschitz randomized transformation, and $\mathcal{M}$ is $(\varepsilon/k, 0)$-DP. By the Lipschitz composition lemma (Lemma 17), Algorithm 2 (`DPTheilSenkMatch`) is $(\varepsilon, 0)$-DP. □

Proofs of Lemmas 7 and 8. The privacy of `DPExpTheilSenkMatch` and `DPWideTheilSenkMatch` follows directly from Theorem 12 and Lemma 5. □

C.2 Sensitivity to Hyperparameter

Choosing optimal hyperparameters is beyond the scope of this work. However, in this section we present some preliminary work exploring the behavior of `DPWideTheilSen` with respect to the choice of θ. In particular, we consider the question of how robust this algorithm is to the setting of the hyperparameter. Figure 8 shows the performance as a function the widening parameter θ on synthetic (Gaussian) data. Note that in each graph both axes are on a log-scale so we see very

[7] Alternatively, to estimate $\tilde{\beta}$, one could compute $\tilde{x}, \tilde{y}$, private estimates of $\bar{x}, \bar{y}$ by adding Laplace noise from $\text{Lap}(0, r_u/n)$ and then compute $\hat{\beta} = \tilde{y} - \hat{\alpha}\tilde{x}$.

large variation in the quality depending on the choice of hyperparameter.

C.3 Pseudo-code for DPExpTheilSen and DPWideTheilSen

In Algorithm 3, we give an efficient method for implementation of the DP median algorithm used as subroutine in DPExpTheilSen, the exponential mechanism for computing medians. To sample efficiently from this distribution, we implement a two-step algorithm following [14]: first, we sample an interval according to the exponential mechanism, and then we will sample an output uniformly at random from that interval. To efficiently sample from the exponential mechanism, we use the fact that sampling from the exponential mechanism is equivalent to choosing the value with maximum utility score after i.i.d. Gumbel-distributed noise has been added to the utility scores [1, 20].

The pseudo-code for DPWideTheilSen is given in Algorithm 5. It is a small variant on Algorithm 3.

Algorithm 3: Exponential Mechanism for Median: $(\varepsilon/k, 0)$-DP Algorithm

Data: $\mathbf{z}$
Privacy params: ε
Hyperparams: n, k, r_l, r_u
$\varepsilon = \varepsilon/k$
Sort $\mathbf{z}$ in increasing order
Clip $\mathbf{z}$ to the range $[r_l, r_u]$
Insert r_l and r_u into $\mathbf{z}$ and set $n = n + 2$
Set maxNoisyScore $= -\infty$
Set argMaxNoisyScore $= -1$
for $i \in [1, n)$ **do**
  logIntervalLength $= \log(\mathbf{z}[i] - \mathbf{z}[i-1])$
  distFromMedian $= \lceil |i - \frac{n}{2}| \rceil$
  score $=$
   logIntervalLength $- \frac{\varepsilon}{2} \cdot$ distFromMedian
  $N \sim$ Gumbel$(0, 1)$
  noisyScore $=$ score $+ N$
  if *noisyScore* $>$ *maxNoisyScore* **then**
    maxNoisyScore $=$ noisyScore
    argMaxNoisyScore $= i$
left $= \mathbf{z}$[argMaxNoisyScore-1]
right $= \mathbf{z}$[argMaxNoisyScore]
Sample $\tilde{m} \sim$ Unif [left, right]
return $\tilde{m}$

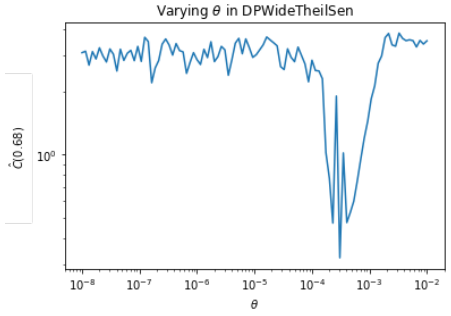


Fig. 8. Experimental results exploring the sensitivity of the hyperparameter choices for DPWideTheilSen. For each dataset $n = 40$, and n datapoints are generated as $x_i \sim \mathcal{N}(0, \sigma^2)$, $y_i = 0.5 * x_i + 0.5 + \mathcal{N}(0, \tau^2)$. The parameters of the data are fixed at $\sigma = 10^{-3}$ and $\tau = 10^{-4}$. The datapoints are then truncated so they belong between 0 and 1. Note that both axes are on a log scale.

Algorithm 4: DPTheilSen: $(\varepsilon, 0)$-DP Algorithm

Data: $\{(x_i, y_i)\}_{i=1}^n \in (\mathbb{R} \times \mathbb{R})^n$
Privacy params: ε
Hyperparams: n, DPmed, hp
$\mathbf{z}^{(\text{p25})}, \mathbf{z}^{(\text{p75})} = [\,]$
for $1 \le i < j \le n$ **do**
  if $(x_j - x_i \neq 0)$ **then**
    $s = (y_j - y_i)/(x_j - x_i)$
    $z_{i,j}^{(p25)} = s\left(0.25 - \frac{x_j + x_i}{2}\right) + \frac{y_j + y_i}{2}$
    $z_{i,j}^{(p75)} = s\left(0.75 - \frac{x_j + x_i}{2}\right) + \frac{y_j + y_i}{2}$
    Append $z_{i,j}^{(p25)}$ to $\mathbf{z}^{(\text{p25})}$ and $z_{i,j}^{(p75)}$ to $\mathbf{z}^{(\text{p75})}$
$\tilde{p}_{25} = \text{DPmed}\left(\mathbf{z}^{(\text{p25})}, \varepsilon/2, (n, \text{hp})\right)$
$\tilde{p}_{75} = \text{DPmed}\left(\mathbf{z}^{(\text{p75})}, \varepsilon/2, (n, \text{hp})\right)$
return $\tilde{p}_{25}, \tilde{p}_{75}$

D DPSSTheilSen

Suppose we are given a dataset $(\mathbf{x}, \mathbf{y})$. Consider a neighboring dataset $(\mathbf{x}', \mathbf{y}')$ that differs from the original dataset in exactly one row. Let $\mathbf{z}$ be the set of point estimates (e.g., the $p25$ or $p75$ point estimates) induced by the dataset $(\mathbf{x}, \mathbf{y})$, and let $\mathbf{z}'$ be the set of point estimates induced by dataset $(\mathbf{x}', \mathbf{y}')$ by Theil-Sen. Formally, for $N = kn/2$, we let $\mathcal{Z}_k : [0,1]^n \times [0,1]^n \to \mathbb{R}^N$ denote the function that transforms a set of point coordinates into estimates for each pair of points. Then $\mathbf{z} = \mathcal{Z}(\mathbf{x}, \mathbf{y})$, $\mathbf{z}' = \mathcal{Z}(\mathbf{x}', \mathbf{y}')$. Notice that changing one datapoint in $(\mathbf{x}, \mathbf{y})$ changes at most k of the point estimates in $\mathbf{z}$. Assume that both $\mathbf{z}$ and $\mathbf{z}'$ are in sorted

Algorithm 5: θ-Widened Exponential Mechanism for Median: $(\varepsilon/k, 0)$-DP Algorithm

Data: $\mathbf{z}$
Privacy params: ε
Hyperparams: n, k, θ, r_l, r_u
$\varepsilon = \varepsilon/k$
Sort $\mathbf{z}$ in increasing order
Clip $\mathbf{z}$ to the range $[r_l, r_u]$
if *n is even* **then**
  Insert m, the true median, into $\mathbf{z}$
  Set $n = n + 1$
for $i \in [0, \lfloor \frac{n}{2} \rfloor]$ **do**
  $z[i] = \max(z_l, z[i] - \theta)$
  $z[n - i - 1] = \min(z_u, z[i] + \theta)$
Insert z_l and z_b into $\mathbf{z}$ and set $n = n + 2$
Set maxNoisyScore $= -\infty$
Set argMaxNoisyScore $= -1$
for $i \in [1, n)$ **do**
  logIntervalLength $= \log(\mathbf{z}[i] - \mathbf{z}[i-1])$
  distFromMedian $= \lceil |i - \frac{n}{2}| \rceil$
  score $=$
   logIntervalLength $- \frac{\varepsilon}{2} \cdot$ distFromMedian
  $N \sim$ Gumbel$(0, 1)$
  noisyScore $=$ score $+ N$
  if *noisyScore* $>$ *maxNoisyScore* **then**
    maxNoisyScore $=$ noisyScore
    argMaxNoisyScore $= i$
left $= \mathbf{z}$[argMaxNoisyScore-1]
right $= \mathbf{z}$[argMaxNoisyScore]
Sample $\tilde{m} \sim$ Unif [left, right]
return $\tilde{m}$

order. Recall the definition of $S^k_{\text{med}\circ\mathcal{Z},t}((\mathbf{x}, \mathbf{y}))$:

$$S^k_{\text{med}\circ\mathcal{Z}_k,t}((\mathbf{x}, \mathbf{y})) = \max\Big\{z_{m+k} - z_m, z_m - z_{m-k}, \max_{l=1,\dots,n} \max_{s=0,\cdots,k(l+1)} e^{-lt}(z_{m+s} - z_{m-(k(l+1)+s)})\Big\},$$

Let $LS^k_{\text{med}}(\mathbf{z}) = \max_{\mathbf{z}' \in \mathbb{R}^N, \text{Ham}(\mathbf{z},\mathbf{z}') \le k} |\text{med}(\mathbf{z}) - \text{med}(\mathbf{z}')|$ be the distance k local sensitivity of the dataset $\mathbf{z}$ with respect to the median. In order to prove that $S^k_{\text{med}\circ\mathcal{Z}_k,t}((\mathbf{x}, \mathbf{y}))$ is a t-smooth upper bound on $LS_{\text{med}\circ\mathcal{Z}_k}$, we will use the observation that

$$LS_{\text{med}\circ\mathcal{Z}_k}(\mathbf{x}, \mathbf{y}) \le LS^k_{\text{med}}(\mathbf{z}).$$

Now Figure 9 outlines the maximal changes we can make to $\mathbf{z}$. For $l \ge 1$ and any interval of $lk + k + 1$ points containing the median, we can move the median to one side of the interval by moving kl points, and to the other side by moving an additional l points. Therefore, for $l \ge 1$,

$$\max_{\mathbf{z}': d(\mathbf{z},\mathbf{z}') \le lk} LS_{\text{med}}(\mathbf{z}') = \max_{s=0,\cdots,lk+k} \{z_{m+s} - z_{m-(lk+k)+s}\} \tag{4}$$

so $S^k_{\text{med},t}(\mathbf{z}) = \max_{l=0,\dots,n} e^{-lt} \max_{\mathbf{z}': d(\mathbf{z},\mathbf{z}') \le lk} LS^k_{\text{med}}(\mathbf{z}')$.

Algorithm 6: Smooth Sensitivity Student's T Noise Addition for Median: $(\varepsilon, 0)$-DP Algorithm

Data: $\mathbf{z}, \{(x_i, y_i)\}_{i=1}^n \in (\mathbb{R} \times \mathbb{R})^n$
Privacy params: ε
Hyperparams: k, n, r_l, r_u, d
Set $t = \frac{\varepsilon}{2(d+1)}$ and $s = \frac{\varepsilon\sqrt{d}}{d+1}$
$S_{\text{median}} = S^k_{\text{med},t}((\mathbf{x}, \mathbf{y}))$
Sample $N \sim$ Student's T(d)
Set $\widetilde{m} = \text{median}(\mathbf{z}) + \frac{1}{s} \cdot S_{\text{median}} \cdot N$
return $\widetilde{m}$

Proof of Lemma 10. We need to show that $S^k_{\text{med}\circ\mathcal{Z}_k,t}((\mathbf{x}, \mathbf{y}))$ is lower bounded by the local sensitivity and that for any dataset $(\mathbf{x}', \mathbf{y}')$ such that $d((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')) \le l$, we have $S^k_{\text{med}\circ\mathcal{Z}_k,t}((\mathbf{x}, \mathbf{y})) \le e^{tl} S^k_{\text{med}\circ\mathcal{Z}_k,t}((\mathbf{x}', \mathbf{y}'))$.

By definition of $S^k_{\text{med},t}$, we see that $S^k_{\text{med}\circ\mathcal{Z}_k,t}((\mathbf{x}, \mathbf{y})) \ge LS^k_{\text{med}}$ (e.g., when $l = 0$ in the formula for $S^k_{\text{med},t}$). Next, we see that

$$S^k_{\text{med},t}(\mathbf{z}) = \max_{l=0,\dots,n} e^{-lt} \max_{\mathbf{z}': d(\mathbf{z},\mathbf{z}') \le lk} LS^k_{\text{med}}(\mathbf{z}') \tag{5}$$

$$\le e^t \cdot \max_{l=1,\dots,n} e^{-lt} \max_{\mathbf{z}'': d(\mathbf{z}',\mathbf{z}'') \le lk} LS^k_{\text{med}}(\mathbf{z}'') \tag{6}$$

$$\le e^t \cdot S^k_{\text{med},t}(\mathbf{z}'), \tag{7}$$

which completes our proof. □

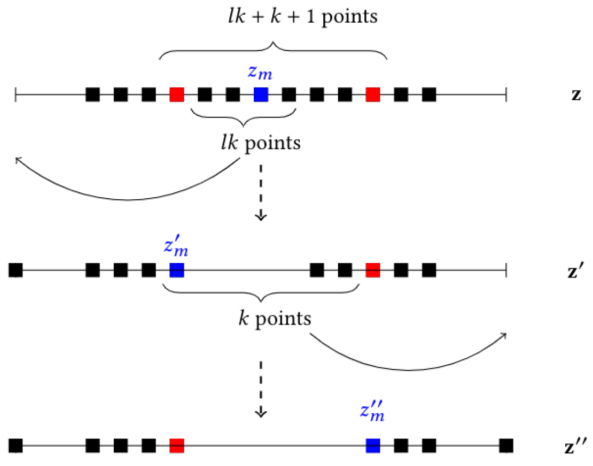


Fig. 9. A brief proof by pictures of Equation 4.

Lemma 20. *Let* $M(\boldsymbol{x}) = median(\boldsymbol{x}) + \frac{1}{s}S^k_{med,t}(\boldsymbol{x}) \cdot N$*, where* N, s *and* t *are computed according to Algorithm 6. Then,* M *is* $(\varepsilon, 0)$*-DP.*

Proof. Let $D_\infty(P||Q) = \sup_{x\in\text{supp}(Q)} \log \frac{p(x)}{q(x)}$ denote the max-divergence for distributions P and Q. Let N be a random variable sampled from StudentsT(d), where $d > 0$ is the degrees of freedom. From Theorem 31 in [9], we have that for $s, t > 0$,

$$\left.\begin{matrix} D_\infty(N||e^t N + s) \\ D_\infty(e^t N + s||N) \end{matrix}\right\} \leq |t|(d+1) + |s| \cdot \frac{d+1}{2\sqrt{d}}$$

The parameters s and t correspond to the translation (shifting) and dilation (scaling) of the StudentsT(d) distribution.

Setting $s = 2\sqrt{d}\left(\frac{\varepsilon' - |t|(d+1)}{d+1}\right)$ as in Algorithm 6, we have that for $|t|(d+1) < \varepsilon'$,

$$\begin{cases} D_\infty(N||e^t N + s) \\ D_\infty(e^t N + s||N) \end{cases} \leq \varepsilon \tag{8}$$

If Equation 8 is satisfied, then by Theorem 46 in [9], the mechanism in Algorithm 6, $M(\mathbf{z}) = \text{median}(\mathbf{z}) + \frac{1}{s}S^t_{\text{median}(\cdot)}(\mathbf{z}) + N$, is $(\varepsilon, 0)$-DP. □

E DPGradDescent

There are three main versions of `DPGradDescent` we consider: (1) `DPGDPure`: $(\varepsilon, 0)$-DP; (2) `DPGDApprox`: (ε, δ)-DP; and (3) `DPGDzCDP`: $(\varepsilon^2/2)$-zCDP. The three algorithms are each given by an instantiation of Algorithm 7. As with traditional gradient descent, there are several choices that have been made in designing this algorithm: the step size, the batch size for the gradients, how many of the estimates are averaged to make our final estimate, how the privacy budget is distributed. We have included this pseudo-code for completeness to show the choices that were made in our experiments. We do not claim to have extensively explored the suite of parameter choices, and it is possible that a better choice of these parameters would result in a better performing algorithm. Differentially private gradient descent has received a lot of attention in the literature. For a more in-depth discussion of DP gradient descent see [6].

Algorithm 7: `DPGD` outline

Data: $\{(x_i, y_i)\}_{i=1}^n \in (\mathbb{R} \times \mathbb{R})^n$
Privacy params: ζ
Hyperparams: $n, T, \tau, \tilde{p}^0_{25}, \tilde{p}^0_{75}$
for $t = 0 : T-1$ **do**
  $\zeta_t = \zeta/T$
  for $i = 1 : n$ **do**
    $\tilde{y} = 2(\tilde{p}^t_{25} * (3/4 - x_i) + \tilde{p}^t_{75}(x_i - 1/4))$
    $\Delta_{i,t} = \begin{pmatrix} [2(y_i - \tilde{y})(3/4 - x_i)]^\tau_{-\tau}, \\ [2(y_i - \tilde{y})(x_i - 1/4)]^\tau_{-\tau} \end{pmatrix}$
  Update Option 1:
    $\Delta_t = \sum_{i=1}^n \Delta_{i,t} + \text{Lap}_2\,(0, 4\tau/\zeta_t)$
  Update Option 2:
    $\Delta_t = \sum_{i=1}^n \Delta_{i,t} + \mathcal{N}_2\left(0, (2\tau/\sqrt{\zeta_t})^2\right)$
  $\gamma_t = \frac{1}{\sqrt{\sum_{l=0}^t \Delta_l^2}}$
  $[\tilde{p}^{t+1}_{25}, \tilde{p}^{t+1}_{75}] = [\tilde{p}^t_{25}, \tilde{p}^t_{75}] - \gamma_t * \Delta_t$
return $\frac{2}{T}\sum_{t=T/2}^T [\tilde{p}^t_{25}, \tilde{p}^t_{75}]$

Lemma 21. *For any* $\rho > 0$*, Algorithm 7 with* $\zeta = \rho$ *and using update option 2 (*`DPGDzCDP`*) is* ρ*-zCDP. For any* $\varepsilon > 0$*, Algorithm 7 with* $\zeta = \varepsilon$ *and using update option 1 (*`DPPure`*) is* $(\varepsilon, 0)$*-DP. For any* $\delta \in (0, 1]$ *and any* $\rho > 0$*, Algorithm 7 with* $\zeta = \rho$ *and using update option 2 is* (ε, δ)*-DP where* $\varepsilon = \rho + \sqrt{4\rho \log\left(\frac{\sqrt{\pi\rho}}{\delta}\right)}$.

Proof. The proof of the first statement is a routine application of Proposition 1.6 in [8], and the proof of the second statement is a routine application of Theorem 11 and Theorem 14. □