Adaptive algorithm in differential privacy: comparative analysis of pre-processing methods

Eero Kalaja

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Mathematics and Statistics

Tekijä — Författare — Author
Eero Kalaja

Työn nimi — Arbetets titel — Title

Adaptive algorithm in differential privacy: comparative analysis of pre-processing methods

Oppiaine — Läroämne — Subject
Mathematics

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's Thesis	June 2020	41 pages

Tiivistelmä — Referat — Abstract

Nowadays the amount of data collected on individuals is massive. Making this data more available to data scientists could be tremendously beneficial in a wide range of fields. Sharing data is not a trivial matter as it may expose indivuals to maliciuous attacks. The concept of differential privacy was first introduced in the seminal work by Cynthia Dwork (2006b). It offers solutions for tackling this problem. Applying random noise to the shared statistics protects the individuals while allowing data analysts to use the data to improve predictions.

Input perturbation technique is a simple version of privatizing data, which adds noise to whole data. This thesis studies an output perturbation technique, where the calculations are done with real data, but only sufficient statistics are released. With this method smaller amount of noise is required making the analysis more accurate.

Yu-Xiang Wang (2018) improves the model by introducing an adaptive AdaSSP algorithm to fix the instability issues of the previously used Sufficient Statistics Perturbation (SSP) algorithm. In this thesis we will verify the results shown by Yu-Xiang Wang (2018) and look in to the pre-processing steps more carefully. Yu-Xiang Wang has used some unusual normalization methods especially regarding the sensitivity bounds. We are able show that those had little effect on the results and the AdaSSP algorithm shows its superiority over SSP algorithm also when combined with more common data standardization methods. A small adjustment for the noise levels is suggested for the algorithm to guarantee privacy conditions set by classical Gaussian Mechanism.

We will combine different pre-processing mechanisms with AdaSSP algorithm and show a comparative analysis between them. The results show that *Robust private linear regression* by Honkela et al. (2018) makes significant improvements in predictions with half of the data sets used for testing. The combination of AdaSSP algorithm with *robust private linear regression* often brings us closer to non-private solutions.

Avainsanat — Nyckelord — Keywords
Differential Privacy, Linear Regression, Machine Learning

Säilytyspaikka — Förvaringsställe — Where deposited
Kumpulan tiedekirjasto

Muita tietoja — Övriga uppgifter — Additional information

Supervisor: Antti Honkela

Contents

Chapter 1

Introduction

The amount of data people produce every day is a colossal figure and a good fraction of this is sensitive data collected from individuals whom do not wish their data to be exposed. Medical records would be something people feel especially sensitive about which is easy to understand considering the damage caused if these were accessible by e.g. insurance companies or potential employers. Yet at the same time that data would be extremely valuable for scientific research. The question is how to make use of that data without risking the privacy of the people.

Seminal work in differential privacy written by Dwork et al. (2006b) has provided us answer for this problem. Applying random noise to large data sets creates a sort of plausible deniability for people whom would otherwise consider that sharing their data has a risk of exposing them to a malicious attacker. Differential privacy enables us to infer statistics of the population using old statistical methods e.g. linear regression by Galton (1886), while being uncertain if any specific individual had their records in the data set.

The paper by Dwork et al. (2006b) was the first publication about differential privacy. It laid out the groundwork which is developed even further by various authors to both increase accuracy in the models and to modulate to distinct situations different data requires. In this paper we will shed some light to different methods using differentially private linear regression with Gaussian mechanism by Dwork et al. (2014a) and our focus will be on AdaSSP algorithm introduced by Wang (2018). This is an upgraded version of the Sufficient Statistics Perturbation algorithm (SSP). The adaptability of the algorithm guarantees required level of privacy without the instability issues of the SSP algorithm.

Wang (2018) tested AdaSSP and several other algorithms with 36 commonly used data sets in the UCI Machine learning repository. We will give a hands on example of the usage of this algorithm with four of those data sets and suggest a few improvements regarding the pre-processing of the data. Our main focus in this paper is explaining the AdaSSP algorithm, how it is an improved version the normal SSP algorithm and how

to use this algorithm in practice. We try to emphasize the challenges of often neglected pre-processing steps, where the fitting decisions regarding normalization of the data can remarkably improve the results. When we go through the theoretical background, we notice that the algorithm seems to require slightly larger noise levels than the ones used by Wang (2018) and we suggest a few changes to ensure that privacy bounds are intact.

The rest of the paper is divided into four chapters. In Chapter 2 we go through the basic principles of differential privacy, linear regression and some mathematical theory which will be needed later on. In Chapter 3 we analyze SSP and AdaSSP algorithms and the data normalization challenges we face when using these algorithms and differentially private linear regression in general. Chapter 4 is reserved for visualizing the effects of different normalization and mapping steps on the error levels and how the small alterations in SSP and AdaSSP algorithms change the results. We will conclude our observations in Chapter 5.

Chapter 2

Background

Motivation for creating differentially private models is summarized well by Dwork et al. (2006b): "–, the goal of a privacy-preserving statistical database is to enable the user to learn properties of the population as a whole while protecting the privacy of the individual contributors." Dwork et al. (2006a) have a background in computer science and therefore the discussion is often circling data bases, trusted servers and answers for the queries. We adopt much of the same framework, where we have three parties. We see the user as the data analyst whom is interested in utilizing the population data for research. The instance whom is in possession of the data, later described by Dwork et al. (2014a), a trusted and trustworthy curator who holds the data of individuals in a database. And at last we have the adversary trying exploit data of individuals. In this paper we use only output perturbation Dwork et al. (2006b), where the correct answers are calculated using the exact data available, but only the noisy versions are reported. Therefore we can simplify the model slightly and let go of the talk about data bases and queries. We still have exact data controlled by the curator, whom will share the perturbed statistics of the data for data analysts. Curator needs to adjust noise levels so that we have a certain level of privacy guarantee protecting the individuals against the malicious adversary.

To put differential privacy in to more exact terms, we wish to publish statistics in a fashion where it is difficult to say which of the two very similar data sets D, D' were used to calculate the results. Usually we are interested in defining a specific limit e^{ϵ} that is the ratio between the probabilities of a randomized algorithm $\mathcal{M}$ giving the same result when using these slightly different data sets. Due to the history of differential privacy, data sets D and D' usually differ only with one row. This one row often represents the data of a single person whose contribution to the results we wanted to obscure. A model introduced by Dwork et al. (2006a) where also the size of the data sets differs by one, underlines the situation where an individual has the decision to share his or her data. In another common definition the size of the data set is set before hand and only one row is

replaced with another in the two neighboring data sets. This makes it easier to build the theory in some cases where the size of the data set can make a big difference.

We will use a following notation in the paper. Non-capital letters refer to scalars and bold letters are used for vectors, which are treated as column vectors. Matrices are referred with capital letters where the columns represent different attributes and the rows are individual data points. In a general case for a data set $D \in \mathbb{R}^{n\times(d+1)}$ we will use a separate matrix $X \in \mathbb{R}^{n\times d}$ for the d regressor values with n observed data points. The last column of the data set D we separate as the vector $\boldsymbol{y} \in \mathbb{R}^n$ and we try to predict these values using the values of the matrix X. We use apostrophe to separate identical neighboring data sets D and D' which differ only with a single row. The default norm will be l_2-norm for vectors and a spectral norm for matrices $\|.\|$. We use a "hat" symbol on top of the estimators. For an element-wise product also known as Hadamard product we use $\odot$ notation and for Hadamard inverse which is a point-wise inverse we write a matrix $B = A^{\circ-1}$ if it applies that $b_{i,j} = a_{i,j}^{-1}$ with all indices i and j.

Next we will go through the some definitions before we start using the differentially private models for the UCI data sets.

2.1 Linear regression

Linear regression is one of the most well known statistical methods and there is a vast amount of publications which improve and extend the work originally introduced by Galton (1886). We will not try to prove or show all the theory behind linear regression, but the following sections should work as a reminder for the reader to better understand the usage in context of differential privacy during the following chapters.

2.1.1 Simple linear regression

In simple linear regression model we estimate the relationship between the variable y and the regressor x

$$y = \beta_0 + \beta_1 x + \epsilon,$$

where the regression coefficients β_0 is the intercept term and the β_1 is the slope. The error term ϵ is expected to have zero mean and the variance is unknown. The mean of y is seen to have a linear relationship with x and the error term is just uncorrelated noise.

We are interested in estimating the optimal fit for the regression coefficients to minimize the sum of squares

$$\arg\min_{\beta_0\beta_1} \sum_{i=1}^{n} \left(y - \beta_0 - \beta_1 x_i\right)^2,$$

which will give the least-squares normal equations when partially differentiated with respect β_0 and β_1. Let us use $\bar{x} = \sum_{i=1}^{n} x_i/n$ for the mean value. Now we can denote the corrected sum of squares and cross products with S as

$$S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

and

$$S_{xy} = \sum_{i=1}^{n} y_i (x_i - \bar{x}).$$

The derivation of the normal equations is omitted, but the solutions are given for $\hat{\beta}_0$ and $\hat{\beta}_1$ as

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \tag{2.1}$$

and

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}. \tag{2.2}$$

2.1.2 Multiple linear regression

Multiple linear regression model is a generalization of the simple linear regression to $\mathbb{R}^d$. The theory can be found from nearly all introduction books in to statistics e.g. Rao et al. (1973) and Weisberg (2005). We expect the error term ϵ to be uncorrelated between the observations. We have d regressors and $d + 1$ regression coefficients

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon.$$

Usually the intercept term β_0 is included in the vector $\boldsymbol{\beta} \in \mathbb{R}^{d+1}$. Then also the regressor variables are expressed in a matrix $X \in \mathbb{R}^{n \times d+1}$ with the first column initiated with ones. Then we can express the setup with

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where also the error term $\boldsymbol{\epsilon}$ is n-dimensional vector.

We want to minimize the residual sum of squares (RSS) in the model and $\hat{\boldsymbol{\beta}}$ is the least squares estimate. Again we will omit the proof and only use the result for the ordinary least squares (OLS) estimator

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \boldsymbol{y}, \tag{2.3}$$

which can be found as long as there exists an inverse matrix $(X^T X)^{-1}$. Sufficient statistics perturbation mechanism looked more closely in Section 3.1 will release a perturbed version of these two parts of the OLS estimator. As our focus in paper by Wang (2018), we will be using the same symbol $\hat{\theta}$ for the OLS estimator instead for easier comparison with the work by Wang.

2.1.3 Ridge regression

Ridge regression, which is also known as Tikhonov regularization Kirsch (2011), Yan (2009), is a well known method for solving ill-posed inverse problems. Regularization parameter λ is added to the multiple linear regression model and we want to minimize the loss function

$$L(\boldsymbol{\beta}) = \|\boldsymbol{y} - X\boldsymbol{\beta}\| + \lambda \|\boldsymbol{\beta}\|^2.$$

This is called L_2-regularization since we favor the values of $\boldsymbol{\beta}$ which have a small l_2-norm.

In ADASSP algorithm an adaptive choice for ridge regression parameter λ will be used to stabilize the least squares estimator when the $X^T X$ matrix has small eigen-values and the chance for a nearly singular inverse matrix is high for the perturbed version of the matrix. The algorithm outputs the ridge estimator

$$\hat{\boldsymbol{\beta}}^{\boldsymbol{R}} = (X^T X + \lambda \boldsymbol{I})^{-1} X^T \boldsymbol{y},$$

though perturbed versions of the sufficient statistics will be used for the calculation.

2.2 Differential privacy

Many of the following theoretical concepts are very similar to the ones shown by Dwork et al. (2014a). However, we are focusing on optimization problems which take place in a real vector space and our primary tool will be Ordinary Least Squares (OLS). Therefore the histogram notation used by Dwork et al. (2014a) is not very fitting and we will make some modifications to the expositions when necessary.

Our goal is to build up the theory for sufficient statistics perturbation mechanism to understand the limitations we have to set for the data. This allows us to get a better understanding regarding the choices we have to make especially in pre-processing phase, but also when running the ADASSP algorithm.

2.2.1 l_2-sensitivity

Sensitivity will help us to bound the amount of change one data point can have for the published results. If this amount of change has no maximum value also the noise needed will have arbitrarily large variance.

We will use l_2-sensitivity as Dwork et al. (2014a) to estimate the maximum effect of an d-dimensional data point to the value of the function $f : A \to B$. We are using the add/remove notation for the neighboring data sets D, D' and therefore also our domain of the function f is of different size for matricies X and X'. We define the function f as follows.

Definition 2.4. Given an arbitrary data set $D \in \mathbb{R}^{n\times(d+1)}$ which includes a matrix $X \in \mathbb{R}^{n\times d}$, the domain A of function f will be a following union:

$$A \subseteq \left(\mathbb{R}^{(n-1)\times d} \cup \mathbb{R}^{n\times d} \cup \mathbb{R}^{(n+1)\times d}\right),$$

while the co-domain of f will be in all cases

$$B \subseteq \mathbb{R}^d.$$

Thus we let neighboring matrix to be $X' \in \mathbb{R}^{(n-1)\times d}$ or $X' \in \mathbb{R}^{(n+1)\times d}$.

Later on we will use A to refer the real vector space which is the domain of the function f, where a small change in the size of the domain is allowed. Next we will define the l_2-sensitivity of the function f.

Definition 2.5. Let function $f : A \to B$ be as above, then the sensitivity Δ_f of the function f is

$$\Delta f = \max_{\substack{X,X'\in A \\ dist(X,X')=1}} \|f(X) - f(X')\|_2,$$

where X and X' are neighboring data sets differing only in one row.

These neighboring data sets are identical excluding one additional row of data. This row serves to represent a collection of data of an arbitrary individual whose privacy protection was the motivation of Dwork et al. (2014a) by using differential privacy. For this formula the notation $dist(X, X')$ is for sets to express the edit distance that is the number of differing rows between the sets. In this thesis the edit distance will always be one as we examine only situations where we want to perturb the effect of any arbitrary observation, which means that one of the sets has one extra row of data and all the rest are identical.

Setting up some limits for the domain and the range is a necessity or sensitivity Δf can have arbitrarily large values. First we set radius $r \in \mathbb{R}$ of an origin centered sphere

$B \subset \mathbb{R}^d$ so that for all the data points $\boldsymbol{x} \in \mathbb{R}^d$ it applies that $\|\boldsymbol{x}\| \leq r$. The dimension parameter d is the number of regressor parameters we have in the data. Now we will have an upper bound for the norm of all the possible row vectors in data set X and its neighboring data set X'. We will show later how to enforce this limit in practice, but now we can define an upper bound on how much a single row in a data set can change the output values of the function f.

2.2.2 (ϵ, δ)-differential privacy

It is helpful to imagine $(\epsilon, 0)$-differential privacy as described by Dwork et al. (2014a), where the released statistics are almost similar no matter which neighboring data sets D, D' are used. In contrast with (ϵ, δ)-differential privacy, where there is a marginal probability that some data sets D, D' produce same statistics with very different probabilities. This relaxation in definition gives us higher utility. The definition we use is similar to the one introduced by Dwork et al. (2006a):

Definition 2.6. An algorithm M which maps $(n \times (d+1))$ matricies to a range B is (ϵ, δ)-differentially private if for all neighboring matricies $D, D' \in A$ it applies:

$$\mathbb{P}(M(D) \in \mathcal{S}) \leq e^{\epsilon}\mathbb{P}(M(D') \in \mathcal{S}) + \delta,$$

for all subsets of $\mathcal{S} \subset B$.

2.2.3 Composition theorems for (ϵ, δ)-differential privacy

Composition theorems shown and proven by Dwork et al. (2014a) help us to combine the building blocks of differential privacy to facilitate the design of algorithms which work in more complex situations. The general composition theorem also by Dwork et al. (2014a) for (ϵ, δ)-differentially private algorithms is most relevant for our situation as we need to split our privacy budget for different parameters in the ADASSP algorithm.

Theorem 2.7. General composition theorem by Dwork et al. (2014a):
Let an algorithm $\mathcal{M} : D \to \mathcal{M}_1(D)$ be an ϵ, δ-differential private algorithm and for $k \geq 2$, $\mathcal{M}_k : (D, s_1, \ldots, s_{k-1}) \to \mathcal{M}_k(D, s_1, \ldots, s_{k-1}) \in \mathcal{C}_k$ be (ϵ, δ)- differential private, for all $(s_{k-1}, \ldots, s_1) \in \bigotimes_{j=1}^{k-1} \mathcal{C}_j$. Then for all neighboring D, D' and all $S \subseteq \bigotimes_{j=1}^{k} \mathcal{C}_j$

$$\mathbb{P}((\mathcal{M}_1, \ldots, \mathcal{M}_k) \in S) \leq e^{k\epsilon}\mathbb{P}'((\mathcal{M}_1, \ldots, \mathcal{M}_k) \in S)) + k\delta.$$

2.2.4 Gaussian mechanism

Gaussian mechanism is a method made well known by Dwork et al. (2014b), which uses the l_2-sensitivity to adjust the amount of noise added for each indices of the parameter to be perturbed. As the name suggests, the noise is taken from a normal distribution $\mathcal{N}(0, \sigma^2)$,where the the variance parameter is a function of some other parameters of the model. In the beginning of Chapter 4 we will discuss the later work by Balle and Wang (2018), where *The Analytic Gaussian Mechanism* is introduced. This offers solutions to the bounds we need to set for the privacy parameter ϵ when using the (classical) Gaussian mechanism. Our focus in this paper is on the classical Gaussian mechanism and we follow the footsteps of Dwork et al. (2014a) and open up the proof below with small adjustments to make it hopefully easier for the reader to follow.

Let $f : \mathcal{A} \to B \subset \mathbb{R}^d$ be an arbitrary function with a bounded l_2-sensitivity Δ_f and let the neighboring data sets $D, D' \in \mathcal{A}$. The Gaussian mechanism adds independently drawn random noise distributed as $\mathcal{N}(0, \sigma^2)$ to each of the d output components of $f(D)$.

Definition 2.8. Let $\epsilon \in (0, 1)$ be arbitrary. For $c^2 > 2\ln(1.25/\delta)$, the Gaussian Mechanism with parameter $\sigma \geq c\Delta_f/\epsilon$ is (ϵ, δ)-differentially private.

Proof. We will be first looking a situation where the function f has a co-domain $B \subset \mathbb{R}$. Given the neighboring data sets D, D', we will consider the ratio between probabilities of a randomizing algorithm giving the same result $\mathcal{M}(D) \in S$ with both data sets. We define $\mathcal{M}(D) = f(D) + err$, where the error term is normally distributed. We want to find when the absolute value of the ratio between their density functions is bounded with e^ϵ. Since they both share the same density function with difference only in the value of the mean parameter μ, we can express this as:

$$\left| \ln \frac{e^{(-1/2\sigma^2)x^2}}{e^{(-1/2\sigma^2)(x+\Delta_f)^2}} \right| \leq \epsilon,$$

where we add the sensitivity parameter Δ_f to the mean value of x. Subtracting the sensitivity value, as we do not know which way the mean is shifted by the sensitivity, is also covered due the symmetry of the situation. Now we can modify

$$\begin{aligned} \left| \ln \frac{e^{(-1/2\sigma^2)x^2}}{e^{(-1/2\sigma^2)(x+\Delta_f)^2}} \right| &= \left| \ln e^{(1/2\sigma^2)\left((x+\Delta_f)^2 - x^2\right)} \right| = \left| \frac{1}{2\sigma^2}(2x\Delta_f + \Delta_f^2) \right| \\ &= \frac{\Delta_f}{\sigma^2} \left| x + \frac{\Delta_f}{2} \right| \leq \epsilon, \end{aligned}$$

where on the last row we assume $\Delta_f > 0$. This gives us

$$\begin{cases} x \leq \dfrac{\sigma^2\epsilon}{\Delta_f} - \dfrac{\Delta_f}{2} & \text{, when } x \geq -\dfrac{\Delta_f}{2} \\ x \geq -\dfrac{\sigma^2\epsilon}{\Delta_f} - \dfrac{\Delta_f}{2} & \text{, when } x < -\dfrac{\Delta_f}{2}. \end{cases}$$

Now we can focus to the upper equation to get a good enough ϵ dependent bound for the values of x. To ensure privacy loss is bounded by ϵ with a probability at least $1 - \delta$, we require

$$\mathbb{P}[|x| \geq \sigma^2\epsilon/\Delta_f - \Delta_f/2] < \delta,$$

where the absolute value of x concludes the negative values of the x. Due the symmetry of the situation we can halve the δ parameter and limit our scope to

$$\mathbb{P}[x \geq \sigma^2\epsilon/\Delta_f - \Delta_f/2] < \delta/2.$$

Gaussian tail bound gives us an upper bound for the probability of getting a value of $x > t$ as:

$$\mathbb{P}[x > t] \leq \frac{\sigma}{t\sqrt{2\pi}} e^{-t^2/2\sigma^2}.$$

With the help of the tail bound we can guarantee (ϵ, δ)-differential privacy as long as

$$\begin{aligned} & \frac{\sigma}{t\sqrt{2\pi}} e^{-t^2/2\sigma^2} < \frac{\delta}{2} \\ \Leftrightarrow\ & \frac{\sigma}{t} e^{-t^2/2\sigma^2} < \frac{\sqrt{2\pi}\delta}{2} \\ \Leftrightarrow\ & \frac{t}{\sigma} e^{t^2/2\sigma^2} > \frac{2}{\sqrt{2\pi}\delta} \\ \Leftrightarrow\ & \ln\Big(\frac{t}{\sigma}\Big) + t^2/2\sigma^2 > \ln\Big(\frac{2}{\sqrt{2\pi}\delta}\Big). \end{aligned}$$

Now by replacing the t with the bound found earlier $t = \sigma^2\epsilon/\Delta_f - \Delta_f/2$, we get

$$\begin{aligned} & \ln\Big(\frac{\sigma^2\epsilon/\Delta_f - \Delta_f/2}{\sigma}\Big) + \frac{(\sigma^2\epsilon/\Delta_f - \Delta_f/2)^2}{2\sigma^2} > \ln\Big(\frac{2}{\sqrt{2\pi}\delta}\Big) \\ \Leftrightarrow\ & \ln\Big(\frac{\sigma^2\epsilon/\Delta_f - \Delta_f/2}{\sigma}\Big) + \frac{(\sigma^2\epsilon/\Delta_f - \Delta_f/2)^2}{2\sigma^2} > \ln\Big(\frac{\sqrt{2}}{\delta\sqrt{\pi}}\Big). \end{aligned} \tag{2.9}$$

Next we need to express the standard deviation parameter σ with the help of the sensitivity Δ_f and privacy budget ϵ. We introduce a new scalar c to scale up the noise we need to add and express standard deviation as $\sigma = c\Delta_f/\epsilon$. It is clear from the previous equation that the noise required is directly proportional to the sensitivity of the function and inversely proportional to the privacy budget ϵ. Now we want to find a bound for c which realizes the inequality in Equation 2.9 and we start by making sure that the first term will be non-negative by expressing

$$\begin{aligned}
&\ln\Big(\frac{\sigma^2\epsilon/\Delta_f - \Delta_f/2}{\sigma}\Big) > 0 \\
\Leftrightarrow\ &\frac{1}{\sigma}\Big(\frac{\sigma^2\epsilon}{\Delta_f} - \frac{\Delta_f}{2}\Big) > 1 \\
\Leftrightarrow\ &\frac{\epsilon}{c\Delta_f}\Big(\frac{(c\Delta_f/\epsilon)^2\epsilon}{\Delta_f} - \frac{\Delta_f}{2}\Big) > 1 \\
\Leftrightarrow\ &\Big(c - \frac{\epsilon}{2c}\Big) > 1 \\
\Leftrightarrow\ &c > \frac{\epsilon}{2c} + 1 \\
\Rightarrow\ &c > \frac{1}{2} + 1 = \frac{3}{2}, \qquad (2.10)
\end{aligned}$$

where the last row can be estimated when we know the bounds $\epsilon \in (0, 1)$ and $c \geq 1$. The upper bound we set for ϵ is something we will consider in Chapter 3. Now it suffices to show that the second part of Equation 2.9 holds the inequality with some value of $c > 3/2$. Again we substitute parameter σ in inequality

$$\frac{(\sigma^2\epsilon/\Delta_f - \Delta_f/2)^2}{2\sigma^2} > \ln\Big(\frac{\sqrt{2}}{\delta\sqrt{\pi}}\Big)$$

and we get

$$\begin{aligned}
\frac{(\sigma^2\epsilon/\Delta_f - \Delta_f/2)^2}{2\sigma^2} &= \frac{1}{2\sigma^2}\Big(\frac{\sigma^2\epsilon}{\Delta_f} - \frac{\Delta_f}{2}\Big)^2 \\
&= \frac{\epsilon^2}{2c^2\Delta_f^2}\Big(\frac{c^2\Delta_f^2\epsilon}{\epsilon^2\Delta_f} - \frac{\Delta_f}{2}\Big)^2 \\
&= \frac{\epsilon^2}{2c^2\Delta_f^2}\Big(\frac{c^4\Delta_f^2}{\epsilon^2} - 2\frac{c^2\Delta_f^2}{2\epsilon} + \frac{\Delta_f^2}{4}\Big) \\
&= \frac{1}{2}\Big(c^2 - \epsilon + \frac{\epsilon^2}{4c^2}\Big) \\
&= \frac{1}{2}\Big(c^2 - \epsilon + \frac{\epsilon^2}{4c^2}\Big) > \ln\Big(\frac{\sqrt{2}}{\delta\sqrt{\pi}}\Big).
\end{aligned}$$

We notice that when $\epsilon \in (0,1)$ the left side of the inequality has a positive derivative with respect to c always when $c > 3/2$ and we get

$$\frac{1}{2}\Big(c^2 - \epsilon + \frac{\epsilon^2}{4c^2}\Big) > \frac{1}{2}\Big(c^2 - 1 + \frac{1}{4(3/2)^2}\Big) > \ln\Big(\frac{\sqrt{2}}{\delta\sqrt{\pi}}\Big).$$

Now we can get a lower bound for c which ensures the value of $\sigma = c\Delta_f/\epsilon$ will be large enough to hold the (ϵ, δ)-differential privacy. That is when

$$\begin{aligned} &\frac{1}{2}(c^2 - 8/9) > \ln\Big(\frac{\sqrt{2}}{\delta\sqrt{\pi}}\Big) \\ \Leftrightarrow\ &(c^2 - 8/9) > 2\ln\Big(\frac{\sqrt{2}}{\delta\sqrt{\pi}}\Big) \\ \Leftrightarrow\ &c^2 > \ln(2/\pi) + \ln e^{8/9} + 2\ln(1/\delta). \end{aligned}$$

We can present this with one real value scalar when we notice that $(2/\pi)e^{8/9} < 1.55 < 1.25^2$ and then it follows that

$$c^2 > 2\ln(1.25/\delta) > \ln(1.55) + 2\ln(1/\delta) > \ln(2/\pi) + \ln e^{8/9} + 2\ln(1/\delta).$$

Now we have all the pieces we need to show that choosing a $c^2 > 2\log(1.25/\delta)$ for the standard deviation parameter $\sigma = c\Delta_f/\epsilon$, we can guarantee (ϵ, δ)-differential privacy. We divide the $\mathbb{R} = R_1 \cup R_2$, where $R_1 = \{f(D) + z : |z| < \sigma^2\epsilon/\Delta_f - \Delta_f/2\}$ and $R_2 = \{f(D) + z : |z| \geq \sigma^2\epsilon/\Delta_f - \Delta_f/2\}$. Now for an arbitrary S we can estimate the probability

$$\begin{aligned} \mathop{\mathbb{P}}_{x\sim\mathcal{N}(0,\sigma^2)}[f(D) + x \in S\ ] &= \mathop{\mathbb{P}}_{x\sim\mathcal{N}(0,\sigma^2)}[f(D) + x \in (S \cap R_1)] + \mathop{\mathbb{P}}_{x\sim\mathcal{N}(0,\sigma^2)}[f(D) + x \in (S \cap R_2)] \\ &\leq e^{\epsilon} \mathop{\mathbb{P}}_{x\sim\mathcal{N}(0,\sigma^2)}[f(D') + x \in (S \cap R_1)] + \delta, \end{aligned}$$

□

The high dimensional case follows conveniently due the spherically symmetric condition and is shown by Dwork et al. (2014a). Gaussian Mechanism can be also used for covariance matricies in the form of Algorithm 1 as shown also by Dwork et al. (2014b). We set the l_2-sensitivity $\Delta_f = 1$ and now $\sigma = \sqrt{2\ln(1.25/\delta)}/\epsilon$.

Algorithm 1 The Gaussian Mechanism: releasing the covariance matrix

Input: Matrix $X \in \mathbb{R}^{n \times d}$, and privacy parameters $\epsilon, \delta > 0$.

Let $E \in \mathbb{R}^{d \times d}$ be a symmetric matrix where the upper triangle (including the diagonal) is i.i.d. samples from $\mathcal{N}(0, \sigma^2)$, and each lower triangle entry is copied from its upper triangle counterpart.

Output: $\widehat{X^T X} \leftarrow X^T X + E$

Chapter 3

Methods

In this chapter we will go through the methods Wang (2018) uses for the 36 different UCI data sets. His main results are two different adaptive algorithms AdaSSP and AdaOPS which he compares to other popular differentially private algorithms. We will analyze the AdaSSP algorithm and consider the pre-processing methods Wang (2018) uses for normalizing the data. We will not use all the 36 different data sets for testing and limit our scope to four of these data sets to keep the amount of work feasible. We chose these four data sets so that their size and shape give a good representation of the different data used to test AdaSSP algorithm earlier. We use the same notation for the data as Wang (2018), where all the data sets are read in as a design matrix $X \in \mathbb{R}^{n \times d}$ with a response vector $\boldsymbol{y} \in \mathbb{R}^n$, where n is the number of data points and d is the number of explanatory variables.

We offer different strategies for normalizing the data and for achieving bounded sensitivity for the regressor and response variables. To improve predictions we suggest some methods which have been often used in the pre-processing phase and in Chapter 4 we will visualize the effects when applied to these four UCI data sets.

3.1 Sufficient statisticis perturbation SSP

The key idea of the sufficient statistics perturbation shown by Vu and Slavkovic (2009), Wang (2018) is to add noise to the released statistic parameters which are required for analyzing the data. In our case the parameters in question are $X^T X$ and $X^T \boldsymbol{y}$, which are required to get the OLS estimator.

For the first parameter the noise added is in the shape of a symmetric matrix E_1, where the values of the upper triangular matrix have been taken from normal distribution $\mathcal{N} \sim (0, \sigma^2)$. Wang (2018) seems to suggest that value of $\sigma^2 = 4\|\mathcal{X}\|^4 \log(4/\delta)/\epsilon^2$ guarantees

(ϵ, δ)-differential privacy. In his notation the $\|\mathcal{X}\|$ is a smallest value for the radius of a ball which contains the set $\mathcal{X}$, which gives us the sensitivity $\Delta = \|\mathcal{X}\|^2$ for X^TX matrix. It is not quite clear where the value σ^2 comes from, but when we take a look at the actual code[1] which Wang presents for running the results, it seems like he has already divided the privacy budget for the two parameters X^TX and $X^T\boldsymbol{y}$. Unfortunately this implicates that the value of σ^2 Wang uses is smaller than the one shown in Algorithm 1 and it may be insufficient to ensure differential privacy. We will use the more conservative value for σ^2 parameter as in the article by Dwork et al. (2014a). There seems to be also a small flaw in the code regarding symmetry of the error matrix E_1, and Wang (2018) ends up using a non symmetric error matrix for the X^TX parameter in his results. In Chapter 4 we will check if this has any effect on the results and confirm the instability issues of the SSP algorithm.

3.2 Adaptive choice of λ and AdaSSP algorithm

One of the main contribution of Wang (2018) was to introduce the adaptive Algorithm 2 for sufficient statistics perturbation. In this section we will briefly explain the idea behind the heuristics Wang (2018) has come up with and offer a small fix for the noise levels to match the (classical) Gaussian Mechanism bounds that guarantee (ϵ, δ)-differential privacy.

Algorithm 2 AdaSSP : Sufficient statistics perturbation with adaptive damping

Input: Data $X, \boldsymbol{y}$. Privacy budget: ϵ, δ, Bounds: $\|\mathcal{X}\|, \|\mathcal{Y}\|$.

1: Calculate the minimum eigenvalue $\lambda_{\min}(X^TX)$.

2: Privately release $\tilde{\lambda}_{\min} = \max\left\{\lambda_{\min} + \frac{\sqrt{\log(6/\delta)}}{\epsilon/3}\|\mathcal{X}\|^2 Z - \frac{\log(6/\delta)}{\epsilon/3}\|\mathcal{X}\|^2, 0\right\}$, where $Z \sim \mathcal{N}(0,1)$.

3: Set $\lambda = \max\{0, \frac{\sqrt{d\log(6/\delta)\log(2d^2/\rho)}\|\mathcal{X}\|^2}{\epsilon/3} - \tilde{\lambda}_{\min}\}$

4: Privately release $\widehat{X^TX} = X^TX + \frac{\sqrt{\log(6/\delta)}\|\mathcal{X}\|^2}{\epsilon/3}Z$ for $Z \in \mathbb{R}^{d\times d}$ is a symmetric matrix and every element from upper triangular matrix is sampled from $\mathcal{N}(0,1)$.

5: Privately release $\widehat{X\boldsymbol{y}} = X\boldsymbol{y} + \frac{\sqrt{\log(6/\delta)}\|\mathcal{X}\|\|\mathcal{Y}\|}{\epsilon/3}Z$ for $Z \sim \mathcal{N}(0, I_d)$.

Output: $\tilde{\theta} = (\widehat{X^TX} + \lambda I)^{-1}\widehat{X\boldsymbol{y}}$

[1] (https://github.com/yuxiangw/optimal_dp_linear_regression)

3.2.1 Adaptive choice of λ

As pointed out also by Wang (2018), the SSP model is quite unstable and may fail arbitrarily bad. The solution he comes up with in algorithm ADASSP is introducing adaptive choice for the ridge regression regularization parameter λ, which is chosen according to the other parameters of data.

The purpose of using ridge regression and the λ parameter is to prevent the noise matrix E_1 from turning the perturbed matrix $\widehat{X^TX} = X^TX + E_1$ in to a nearly singular matrix. Wang (2018) approaches the issue in by choosing the value for λ in a way that $\|E_1\| \leq (\lambda_{\min}(X^TX) + \lambda)/2$ is a high probability event. At the same time λ needs to hold the upper bound Wang found for $F(\hat{\theta}_\lambda) - F(\theta^*)$, where $F(\theta^*)$ and $F(\hat{\theta}_\lambda)$ are the RSS estimates with the optimal choices of θ for least squares solution and ridge regression objective. Wang comes up with an upper bound

$$F(\hat{\theta}_\lambda) - F(\theta^*) = O\left(\frac{d\|\mathcal{X}\|^2(\|\mathcal{Y}\|^2 + \|\mathcal{X}\|^2\|\theta^*\|^2)\log(6/\delta)\log(2d^2/\rho)}{(\lambda + \lambda_{\min})\epsilon^2}\right),$$

which behaves nicely even with $\lambda_{\min} = 0$ when λ has values bound by

$$\lambda = \Theta\left(\sqrt{d\log(6/\delta)\log(2d^2/\rho)}\left(\frac{\|\mathcal{X}\|\|\mathcal{Y}\|}{\|\theta^*\|} + \|\mathcal{X}\|^2\right)/\epsilon\right). \tag{3.1}$$

Therefore Wang (2018) uses simple heuristic to bind the value to the bounds above by suggesting

$$\lambda = \max\left\{0, \frac{\sqrt{d\log(6/\delta)\log(2d^2/\rho)}\|\mathcal{X}\|^2}{\epsilon/3} - \lambda^*_{\min}, \right\} \tag{3.2}$$

where $\lambda^*_{\min}$ is a differentially private high probability lower bound of the $\lambda_{\min}$ and is given as

$$\lambda^*_{\min} = \max\left\{\lambda_{\min} + \frac{\sqrt{\log(6/\delta)}}{\epsilon/3}\|\mathcal{X}\|^2 Z - \frac{\log(6/\delta)}{\epsilon/3}\|\mathcal{X}\|^2, 0\right\}.$$

Therefore the Algorithm 2 offers large enough value for the parameter λ even when the smallest eigenvalue $\lambda_{\min}(X^TX)$ is too small to guarantee robust regression estimate. Yet it seems capable to hold the upper bound for the difference between RSS estimates for the least squares solution θ^* and the ridge regression objective $\hat{\theta}_\lambda$.

3.2.2 Fixing the noise levels

In the ADASSP algorithm the privacy budget has been divided to three parameters $\lambda_{\min}, X^TX$ and X^Ty. Unfortunately Wang (2018) uses the Gaussian mechanism again

with the same scalar value as mentioned in Section 3.1, which we have not been able to verify sufficient. Therefore we modify the algorithm to see how much the results change when the added noise has larger variance as in Algorithm 1 by Dwork et al. (2014b), where $\sigma = \sqrt{2\ln(1.25/\delta)}/\epsilon$. We write the scalar $2\log\big(1.25/(\delta/3)\big)$ as $2\log\big(3.75/\delta\big)$ and present the new version of the Algorithm 3.

Algorithm 3 Updated ADASSP : Fixed noise levels

Input: Data $X, \boldsymbol{y}$. Privacy budget: ϵ, δ, Bounds: $\|\mathcal{X}\|, \|\mathcal{Y}\|$.

1: Calculate the minimum eigenvalue $\lambda_{\min}(X^TX)$.

2: Privately release $\tilde{\lambda}_{\min} = \max\Big\{\lambda_{\min} + \frac{\sqrt{2\log(3.75/\delta)}}{\epsilon/3}\|\mathcal{X}\|^2 Z - \frac{2\log(3.75/\delta)}{\epsilon/3}\|\mathcal{X}\|^2, 0\Big\}$, where $Z \sim \mathcal{N}(0,1)$.

3: Set $\lambda = \max\{0, \frac{\sqrt{d2\log(3.75/\delta)\log(2d^2/\rho)}\|\mathcal{X}\|^2}{\epsilon/3} - \tilde{\lambda}_{\min}\}$

4: Privately release $\widehat{X^TX} = X^TX + \frac{\sqrt{2\log(3.75/\delta)}\|\mathcal{X}\|^2}{\epsilon/3}Z$ for $Z \in \mathbb{R}^{d\times d}$ is a symmetric matrix and every element from upper triangular matrix is sampled from $\mathcal{N}(0,1)$.

5: Privately release $\widehat{X\boldsymbol{y}} = X\boldsymbol{y} + \frac{\sqrt{2\log(3.75/\delta)}\|\mathcal{X}\|\|\mathcal{Y}\|}{\epsilon/3}Z$ for $Z \sim \mathcal{N}(0, I_d)$.

Output: $\tilde{\theta} = (\widehat{X^TX} + \lambda I)^{-1}\widehat{X\boldsymbol{y}}$

3.3 Practicalities of differentially private OLS

Pre-processing steps get often little attention as they are seen such standard procedures, but we wish to shed some light on them and the options we have available. When working with data sets it is common to standardize the data column wise to zero mean and unit variance before solving the optimal value for the OLS estimator $\hat{\boldsymbol{\theta}}$. With non-private models we would have the same mean values and standard deviations (sd) of the column vectors to use for training and test sets. Differential privacy complicates the situation slightly as we usually need to share our privacy budget to all published parameters and take in to account the constraint caused by sensitivity during normalization process to achieve tight bound for the minimum and maximum values of the data.

The curator of the data set whom is about to share the necessary values for the $\hat{\boldsymbol{\theta}}$ estimate is left with three options on how to proceed with the other statistics required for normalizing the data. (1) Curator hopes that the data analyst using the released values has large enough data set available to be able to calculate reasonable estimates for the column mean and sd values without access to the training data. (2) The additional statistics are released without perturbation if this is considered an acceptable risk. (3) Or the perturbed version of the additional statistics are released and some privacy budget is

spent on them. We will continue with the second strategy as we find it most fitting with the decisions made by Wang (2018) and with the row specific norm mapping we discuss next.

Wang (2018) does also normalize the data matrix $X \in \mathbb{R}^{n \times d}$ column wise to zero mean and unit variance, but he uses row norms for mapping the values to a unit sphere as shown in Section 3.3.1. The values of the response vector $\boldsymbol{y}$ are divided with the $y_{\max}$. The mapping to unit sphere or some bounded subspace of $\mathbb{R}^d$ is a vital part of the pre-processing which ensures that the sensitivity can be limited to a small enough constant. The row norms of matrix X that Wang (2018) uses for mapping makes the model rather peculiar and even more so when he does not divide the values of the response vector $\boldsymbol{y}$ with the corresponding row norms of the matrix X. Row operations which take parameters from the row values are likely to add correlated noise. We will briefly look in to extending the row operations of the data matrix X to the corresponding values of the response vector $\boldsymbol{y}$ in Section 3.3.1 and examine the benefits and losses.

As an alternative approach from Wang, we use a maximum norm of the data points to create a more robust mapping to a unit sphere. Our main focus will be with this model and it is explained in more detail in Section 3.3.2. We also show the possibility of releasing unstandardized regression coefficients in the last section of this chapter.

3.3.1 Mapping data to unit sphere using row specific norms

As we mentioned above, Wang (2018) uses a row specific norm, where all the row vectors $\boldsymbol{x_i}$ for $i \in [1, \ldots, n]$ of matrix $X \in \mathbb{R}^{n \times d}$ are divided with the row specific norm $\|\boldsymbol{x_i}\|$. This is a different operation from a standard normalization phase done before, where all the values of X have first their column means $\bar{x_j}$ for $j \in [1, \ldots, d]$ subtracted and then divided with column wise standard deviation values sd_j. We use a vector notation for these mean and standard deviation vectors in the following equations with symbols $\boldsymbol{m_X}$ for the vector of means and $\boldsymbol{sd_X}$ for the vector of standard deviations. The sensitivity bound needs to apply to the response vector $\boldsymbol{y}$ also and the vector is divided with maximum value $y_{\max} = \max_{i \in [1,\ldots,n]} \|y_i\|$ of the training set.

Let us consider a test set $X^* \in \mathbb{R}^{m \times d}$ for which we are trying to make the predictions $\boldsymbol{y^*} \in \mathbb{R}^m$ by using our model. From the training data we take given the perturbed $\hat{\boldsymbol{\theta}}$ estimate, maximum value of the response variables $y_{\max}$, a vector of column means $\boldsymbol{m_X} \in \mathbb{R}^d$ and a vector of column standard deviations $\boldsymbol{sd_X} \in \mathbb{R}^d$. Again we will express the values of the test set X^* with the help of row vectors $\boldsymbol{x_i^*}$ for $i \in [1, \ldots, m]$ and it applies that

$$\widehat{y_i^*}/y_{\max} = \left(\frac{(\boldsymbol{x_i^*} - \boldsymbol{m_X}) \odot (\boldsymbol{sd_X})^{\circ -\mathbf{1}}}{\|(\boldsymbol{x_i^*} - \boldsymbol{m_X}) \odot (\boldsymbol{sd_X})^{\circ -\mathbf{1}}\|} \right) \hat{\boldsymbol{\theta}}, \tag{3.3}$$

for all observations $i \in [1, \ldots, m]$ in test set.

An intriguing question arises with this approach. What happens if we extend the division with the row norm also for the response variable y_i^*. In order to keep the sensitivity bounded we need to define a new maximum value from the training data

$$y_{\max_2} = \max_{i \in [1,\ldots,m]} \frac{y_i/y_{\max}}{\|(\boldsymbol{x_i} - \boldsymbol{m_X}) \odot (\boldsymbol{sd_X})^{\circ -\mathbf{1}}\|}.$$

Now we get another model for estimating the response variables, where

$$\frac{\widehat{y_i^*}/y_{\max}}{\|(\boldsymbol{x_i^*} - \boldsymbol{m_X}) \odot (\boldsymbol{sd_x})^{\circ -\mathbf{1}}\|}/y_{\max_2} = \left(\frac{(\boldsymbol{x_i^*} - \boldsymbol{m_X}) \odot (\boldsymbol{sd_x})^{\circ -\mathbf{1}}}{\|(\boldsymbol{x_i^*} - \boldsymbol{m_X}) \odot (\boldsymbol{sd_x})^{\circ -\mathbf{1}}\|} \right) \hat{\boldsymbol{\theta}} \tag{3.4}$$

for all $i \in [1, \ldots, m]$. We will briefly look in to advantages of Equation 3.4 in Chapter 4. Even though the results seem promising with the row specific norms, we realize that the model is quite different form the standard OLS. Dividing the rows of matrix X with different row based scalars makes little sense in lower dimensions and even less so if it does not affect the response variable. In any case, working with row specific norms could be an interesting approach, but we feel that this model is off the topic for us. Wang (2018) did not use intercept columns either for the data sets which is also clear from the model laid out above. Now we want to keep our focus in the adaptive version of differentially private OLS and we opt to another option to keep our sensitivity bounded and our data in the unit sphere.

3.3.2 Mapping data to unit sphere with maximum row norm

Let us use the same test setting as in previous section with matrix $X^* \in \mathbb{R}^{m \times d}$. Sensitivity needs to be bounded without disturbing the principles of our model. We achieve this by mapping the data points of our matrix ${X^*}^{m \times d}$ to unit sphere by dividing all the values with additional parameter we need from the training set. The maximum row norm of the training data $x_{\max} = \max_{i \in 1,\ldots n} \|\boldsymbol{x_i}\|$. This simplifies the Equation 3.3 and we have now

$$\widehat{y_i^*}/y_{\max} = \left(\frac{(\boldsymbol{x_i^*} - \boldsymbol{m_X}) \odot (\boldsymbol{sd_x})^{\circ -\mathbf{1}}}{x_{\max}} \right) \hat{\boldsymbol{\theta}}, \tag{3.5}$$

for all observations $i \in [1, \ldots, m]$ in test set. The weakness of this approach is that one outlier may force the mapping of all the other data points in to a very small space. Even

though we find this unfortunate, we consider it a reasonable price to pay for having a more robust model. It may feel odd that many of the statistics are taken from the training set without perturbing them first. This could be easily done by splitting the privacy budget for these parameters also by using the composition Theorem 2.7. We want to keep the comparison easy with the results by Wang (2018) and we will leave these as they are in the experiments.

3.3.3 Unstandardized regression coefficients

In the previous section one may have wondered why the normalizing parameters are not included in to the perturbed versions of the sufficient statistics. Normalization steps, subtracting mean and division with standard deviation, are an affine transformation and therefore can be included in the $\hat{\theta}$ estimate. This is definitely an option worth exploring, but we are not able to create a model where this decision could or at least should be included by default. There is no need for the row operations on the matrix X now as we use maximum norm mapping so we can use more simple notation below. We also include the intercept parameter θ_0 which was not part of the model used by Wang (2018). Now for any row vector $\boldsymbol{x_i}$ of $X \in \mathbb{R}^{n \times d}$ we have

$$\widehat{y}_i / y_{\max} = \theta_0 + \sum_{j=1}^{d} \left(\frac{\theta_j (x_{i,j} - \bar{x_j})}{x_{\max} \times sd_j} \right), \tag{3.6}$$

for all $i \in [1, \ldots, n]$ when $x_{\max}$ is the maximum row norm, $y_{\max}$ maximum value of vector $\boldsymbol{y}$ and $\bar{x_j}$ is the column mean of j:th column of matrix X.

If we want to release unstandardized regression coefficients, we can edit the Equation 3.6 in to

$$\begin{aligned}
\widehat{y}_i =& y_{\max} \Big(\theta_0 - \sum_{j=1}^{d} \big(\frac{\theta_j \bar{x_j}}{x_{\max} \times sd_j} \big) + \sum_{j=1}^{d} \big(\frac{\theta_j x_{i,j}}{x_{\max} \times sd_j} \big) \Big) \\
=& y_{\max} \theta_0 - \sum_{j=1}^{d} \big(\frac{y_{\max} \theta_j \bar{x_j}}{x_{\max} \times sd_j} \big) + \sum_{j=1}^{d} \big(\frac{y_{\max} \theta_j x_{i,j}}{x_{\max} \times sd_j} \big).
\end{aligned}$$

Now unstandardized values for the θ parameters are

$$\theta'_j = \frac{y_{\max}}{x_{\max} sd_j} \theta_j$$

$$\theta'_0 = y_{\max} \theta_0 - \sum_{j=1}^{d} \theta'_j \bar{x_j}$$

and the model becomes

$$\widehat{y}_i = \theta'_0 + \sum_{j=1}^{d} \theta'_j x_{i,j}, \tag{3.7}$$

for all $j \in [1, \dots n]$. In some cases this will be a good design for releasing the perturbed sufficient statistics with normalizing parameters included in them. However, if those additional statistics need to be also perturbed before releasing, additional work will be needed when estimating the privacy conditions. In the Equation 3.7 we are about to release the values $y_{\max}, x_{\max}, \bar{x_j}$ and sd_j as part of the perturbed θ'_j value without parameter specific noise tailored for those additional parameters. This requires very subtle consideration on the quality of these additional parameters and is a very data specific decision to make.

As an example we can use the data sets we have been working on in this paper. The sensitivity bounds are very different for the $X^T X$ and $X^T \boldsymbol{y}$ parameters than for the true means and standard deviation values of the data sets. The motivation for the mapping phase was to bring the sensitivity to lower bounds. We require a data set specific analysis to estimate if it makes sense to combine these parameters which have very different sensitivities. In some cases the mean values may be unnecessary for the data analyst, and we end up increasing the noise in the sufficient statistics for nothing.

3.4 Robust private linear regression

Up until now we have only considered different ways of projecting data to subset of $\mathbb{R}^d$ in a way where the data outliers have set the density for pre-processed data. This is particularly problematic with differential privacy, where we try to keep the sensitivity Δ_f quite small. It can result that almost all data points will be projected to extreme vicinity of zero. In this section we explain the idea behind the model *robust private linear regression* by Honkela et al. (2018), which offers a solution for the problem described above. In brief *robust private linear regression* uses clipping on the raw data prior normalizing and mapping phase to bring the outliers to tighter bounds ideally set in co-operation with the experts of research field. By clipping the data in to smaller subset of $\mathbb{R}^d$ also the required noise for data perturbation will be smaller. We have very little prior knowledge of these four UCI data sets we have been working on and therefore variable selection and discarding independent variables is not a realistic option for us. We have neither acquired such understanding of any data set that we would feel comfortable in suggesting exact tighter bounds. As we still wish to try combining the *robust private linear regression* model with ADASSP algorithm, we will instead use a simple heuristic for clipping in Section 4.2.2.

Put in to a form of an equation, we need to add another step to pre-processing before we normalize the data. We will use notation $g_c(X)$ to describe a function which will be clipping the input matrix X column wise to percentiles with $c\sigma$ sigma units. Now for a training set matrix $X \in \mathbb{R}^{n \times d}$ with a vector of column means $\boldsymbol{m_X} \in \mathbb{R}^d$ and column of standard deviations $\boldsymbol{sd_X} \in \mathbb{R}^d$ we have a function

$$g_c(X) = g_c(X_{i,j}) = \left\{ \min(X_{i,j}, \quad \boldsymbol{m_X}_j + c \times \boldsymbol{sd_X}_j) \middle| i \in [1, \ldots, n], j \in [1, \ldots, d] \right\}. \tag{3.8}$$

Honkela et al. (2018) uses the clipping for both minimum and maximum values, but as can be seen in Equation 3.8, we enforce clipping only on the higher end of the data. That is a technical decision we made after noticing that only one of our data sets (elevators) has negative regressor values, but the rest have plenty of values in the near vicinity of zero.

The absolute strength in *robust private linear regression* when evaluated with respect to other pre-processing methods we have seen so far is the simplicity. In the model with our data curator and the data analyst, the curator can use clipping on the data without a need to share any additional information about the process the data analyst. Predictions will often improve for the data analyst as the sufficient statistics parameters are more robust when less noise is needed for perturbation.

Just as a curiosity we try in Figure 4.8 the effect of sharing the clipping parameters also for the data analyst. This is not nearly as lean model with high utility as the one introduced by Honkela et al. (2018), but trying this now is a small cost as we have already been working with models with challenging information sharing between the curator and the data analyst. We can implement this addition with function g_c to Equation 3.5 so for the testing set matrix $X^* \in \mathbb{R}^{m \times d}$ and response vector $\boldsymbol{y^*} \in \mathbb{R}^m$ we have

$$\widehat{y_i^*}/y_{\max} = \left(\frac{(g_c(\boldsymbol{x_i^*}) - \boldsymbol{m_X}) \odot (\boldsymbol{std_x})^{\circ -1}}{x_{\max}} \right) \hat{\boldsymbol{\theta}}, \tag{3.9}$$

for all observations i $[1, \ldots m]$. We will calculate new values for the column means $\boldsymbol{m_X} \in \mathbb{R}^d$, column standard deviations $\boldsymbol{sd_X} \in \mathbb{R}^d$ and $x_{\max}$ values after the clipping phase.

3.5 Evaluating results

Wang (2018) estimates the success of the algorithms under the Gaussian model

$$\boldsymbol{y} = X\boldsymbol{\theta} + \mathcal{N}(0, \sigma^2 I_n),$$

by comparing the results of the differentially private estimates of $\hat{\boldsymbol{y}}$ to the non-private solution $\boldsymbol{y^*}$. The error terms calculated in the graphs are given by equation

$$err = (\boldsymbol{y} - \hat{\boldsymbol{y}})^2/n,$$

where $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]$ are the true values of the response variables in the given data set. One might argue that using a coefficient of determination R^2 with a total sum of squares in the denominator could be a more fitting choice for comparing results, but for the sake of easier comparison with the results by Wang (2018) we will use the same error function.

Chapter 4

Results and discussion

Next we will use the algorithms with different pre-processing stesps on four of the same UCI data sets as in the paper by Wang (2018). The chosen data sets represent very different types of data both in the number of observations and in the number of regressor variables. We want to underline the difficulties of implementing differential privacy in small and moderately sized data sets and choose two out of four from the lower end of the scale with observations, other one of them with only five regressor variables.

To verify the results by Wang (2018), we first translate the Matlab code shared in Github to Python. As our purpose is to make it easier for people to use and understand privatizing methods, we had to make some changes to the scale of some parameters to create a more robust guideline. Especially important point is raised in Zhao et al. (2019) "Reviewing and improving the Gaussian Mechanism", which focuses on magnitude of the privacy budget parameter ϵ. It is also clear from the proof of Definition 2.8 that ϵ should not be given values larger than one when using Algorithm 1 by Dwork et al. (2014b). According to article Zhao et al. (2019), in many scientific articles the algorithms have been tested with larger values of ϵ, which in fact do not always guarantee ϵ, δ-differential privacy. This is also the case for the article of our interest by Wang (2018), with the values of epsilon used being as large as 10. Later work by Balle and Wang (2018) uses a numerical method for calculating optimal values for the variance parameter in Gaussian mechanism. Balle and Wang (2018) have named it *Analytical Gaussian mechanism* and it solves issues with both high privacy regime, where $\epsilon \to 0$ and also low privacy regime when $\epsilon > 1$. Instead of estimating the probabilities with the help of Gaussian tail bounds, they calculate the cumulative distribution function values for the normal distribution numerically. However, the work by Wang (2018) follows classical Gaussian Mechanism and we make small adjustments to the parameters so it will stay within the bounds set by Algorithm 1.

As we are not able to use quite as high values for the privacy budget ϵ as 10, we can

use higher values than one. When we split the privacy budget in AdaSSP algorithm for the three parameters $X^TX, X^T\boldsymbol{y}$ and $\lambda_{\min}$ with the composition Theorem 2.7, we use the Gaussian mechanism with privacy budget values $\epsilon/3$ and $\delta/3$. This enables us to use an ϵ three times larger than the limit set by the Gaussian mechanism and for most of the figures we will be testing values of ϵ as high as three. The only exceptions we have with the first figures where we demonstrate the instability issues of the SSP algorithm, which splits the privacy budget for only two parameters $X^TX, X^T\boldsymbol{y}$, setting us an upper limit $\epsilon = 2$.

As our goal has been making an easily approachable example of using differential privacy in the context of linear regression, we will next see the benefits of applying some standard pre-processing methods for the data. It is also interesting to see if these methods have strong differences in the error term between the two algorithms SSP and AdaSSP. As we wish to use adjusted version of the AdaSSP Algorithm 3, we will first show that this gives similar results with the original Algorithm 2. Even thought the results are not as good with Algorithm 3, we can be more confident now that the privacy bounds are met.

Another change we wish to make is tracking the pre-processing normalization steps and the sensitivity bound mapping in order to give a more realistic estimates for the unseen data. This will also weaken the results slightly compared to Wang's approach when we use training set statistics to normalize test sets, instead of normalizing everything with same parameters in the beginning. However, this should emphasize all the practicalities one needs to remember when working with real data using differential privacy. It is also more authentic way to test the estimation errors when the test set has not been given pre-processed.

4.1 Algorithm alterations and unit sphere mapping

4.1.1 Instability of the SSP algorithm

First as a motivation we run the SSP algorithm on the four UCI regression data sets with the same pre-processing with Wang (2018) in Figure 4.1. We also add a fixed version of the algorithm where the noise added to X^TX matrix is symmetric and the variance parameter for the noise is changed to follow Algorithm 1. It is clear that the SSP algorithm can be very unstable as a result of near singular inverse matrix $\widehat{X^TX}$. Fixing the noise symmetric for this matrix does little to change the matter and the error function seems to often get arbitrarily large values. We have included in the Figure 4.1 a non-private version of the ridge regression with $\lambda = 1$ and also the trivial solution $\boldsymbol{\theta} = \overrightarrow{0} \in \mathbb{R}^d$ in the same fashion as Wang (2018) had in his results.

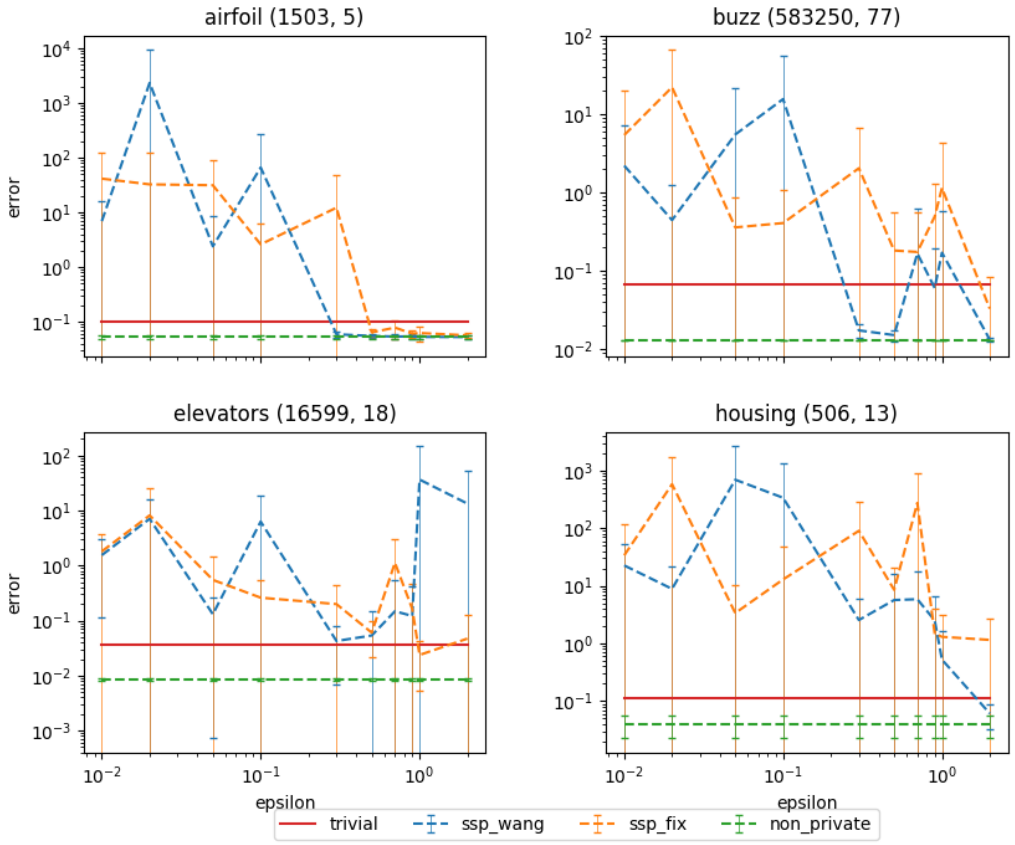

Figure 4.1: Instability of the SSP algorithm visualized. It is clear that the SSP algorithm by Wang (2018) (dashed blue line) and the one with symmetric noise matrix (dashed orange line) have trouble to match the error levels of the trivial solution (red line).

4.1.2 Fixing the noise levels for AdaSSP algorithm

In the following results we want to use the fixed version of the AdaSSP Algorithm 3 where we made slight alterations to the variance parameter of the Gaussian mechanism of Algorithm 2 introduced by Wang (2018). As we were not able to show that Wang's scalars for the variance parameter σ^2 were sufficient, we opt using the Algorithm 3 instead to achieve the ϵ, δ- differential privacy. In the Figure 4.2 we show that both versions of the AdaSSP algorithm give very similar results even though the usage of higher noise levels slightly increases the error level of the fixed Algorithm 3 as expected.

The over all behavior of both AdaSSP algorithms are very similar and we feel confident using mostly the Algorithm 3 of AdaSSP in the following figures where we compare

different pre-processing methods. The same applies for the fixed SSP algorithm with symmetric noise matrix, when we feel that showing the results in comparison with AdaSSP is beneficial. As we will next also change the pre-processing steps, direct comparison with results by Wang (2018) becomes more difficult. By now it should be clear that we get converging results with Wang's using fixed version of the SSP and AdaSSP algorithms if we use the same pre-processing methods. Also AdaSSP algorithm still shows remarkable improvements compared to SSP algorithm, even when slightly larger noise levels for perturbation are used for perturbation than in the paper by Wang (2018).

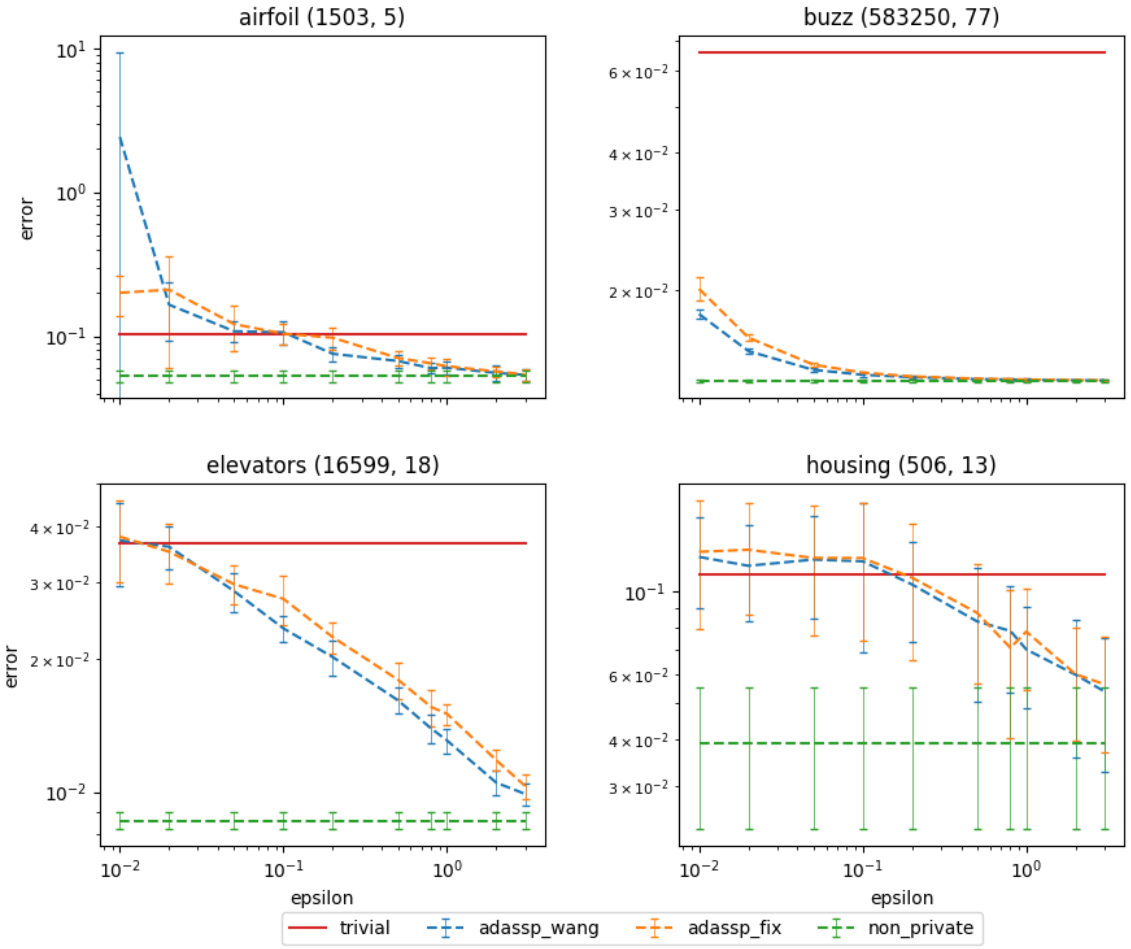

Figure 4.2: AdaSSP with different noise levels for perturbation.

4.1.3 Results for different mapping strategies

In Section 3.3 we discussed the necessity to map the data points in to limited subspace of $\mathbb{R}^d$. When we bound the maximum norm between data points in the data, we are also able achieve bounded sensitivity for the function f, which gives us the statistics of our interests.

In Figure 4.3 we present the differences between using row specific norms and the maximum norm for mapping data to unit sphere. We show the results for three different graphs for both mapping strategies. These are the trivial solution ($\boldsymbol{\theta} = \overrightarrow{\mathbf{0}}$), non-private solution (ridge regression with $\lambda = 1$) and the fixed ADASSP algorithm. The first observation is that even the best mapping strategy for non-private solution depends on the data set and it can not be concluded that one is always superior to another. For ADASSP we use dashed blue line for the Equation 3.3 with row norm mapping and dashed red line for the Equation 3.5 with maximum norm mapping. The row norm mapping seems to give us improving results as the dimensionality of the regressor vectors grows. Considering that the operation would be extremely odd for one dimensional data, where all the regressor values would be forced to have value one, it is something we might expect to happen. Also row specific norms tend to converge towards the same value for standardized data as the number of dimensions grows for the regressor vectors. This suppresses the possible disturbance row specific norm mapping can cause by a single dimension with several outliers or extreme values.

The results of Figure 4.3 show that with these data sets we achieve low sensitivity bounds with better results in prediction with row norm mapping strategy, but as we are not quite certain of the robustness of the model, we choose to focus on the maximum norm method. The maximum norm mapping is extremely sensitive to outliers in data even if the values are problematic only in one dimension. We will show in the following figures the effects of different pre-processing methods when combined with maximum norm mapping strategy.

Before we continue to these maximum norm mapping results, we have Equation 3.4 to test using row specific norms. We compare the results between row norm mapping as in Equation 3.3 to one with the mapping operation extended to effect also the values of response variable $\boldsymbol{y}$ as in Equation 3.4. We differentiate these two in Figure 4.4 by using postfix "_y_included" for graphs with the data pre-processed along Equation 3.4. We use the same set of functions as in the previous example. The extension of the row operation has clearly an effect, but the results are quite mixed. Again it is not even clear if the extended row operation is always beneficial for the non-private solution, since the lower error level depends on data set. This will conclude our analysis with row specific norms and rest of the paper we focus on data with maximum norm mapping used in the pre-processing step. Continuing with the usage of row specific norms could be an

interesting research topic, but that would require a different theoretical approach.

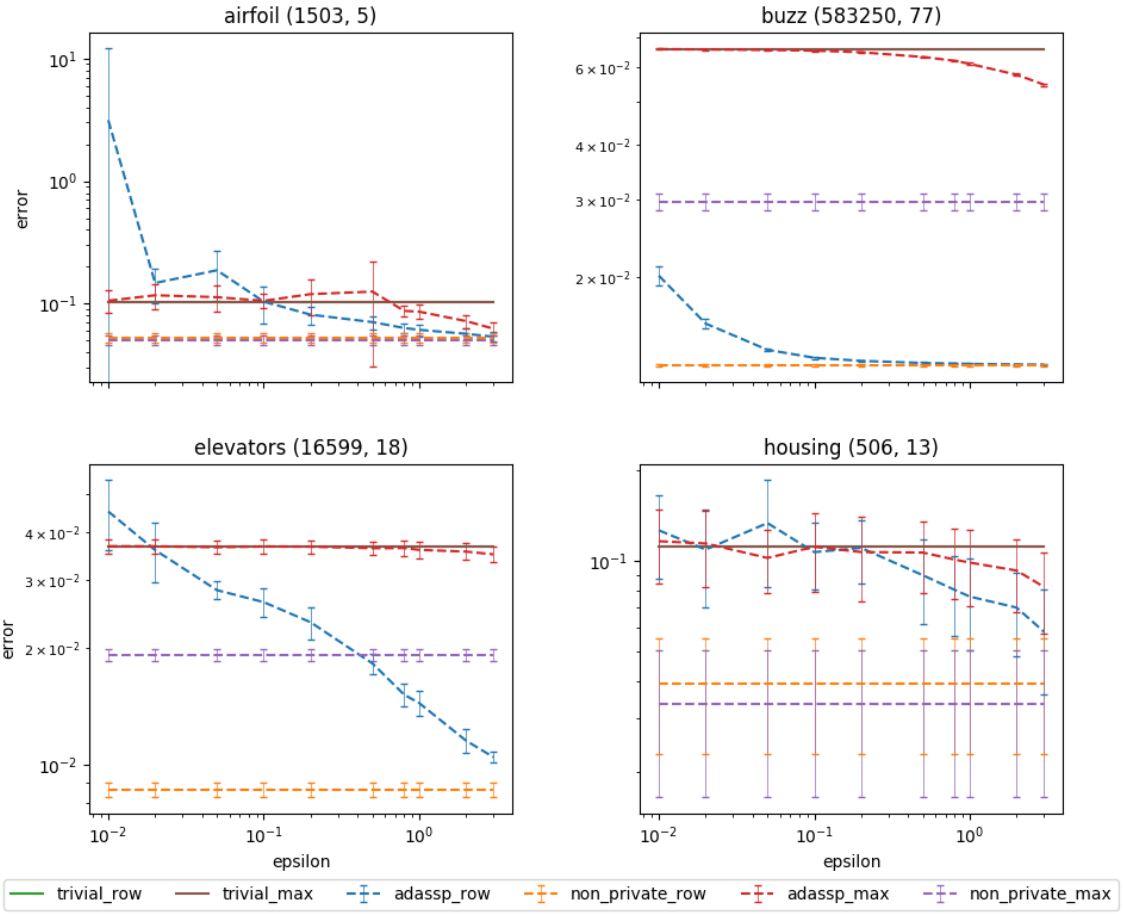

Figure 4.3: Two different methods used for mapping data to unit sphere. We use postfix "_max" when data was mapped with maximum norm value and "_row" when mapping was done with row specific norms.

4.2 Tracking the pre-processing steps

As we have now come to a conclusion with the versions of algorithms to use and the mapping strategy to implement for bounded sensitivity, we are ready focus on the more practical decisions. We also wish to be able to track down how the changes we make in the pre-processing steps effect the results when user of the released statistics, to whom we

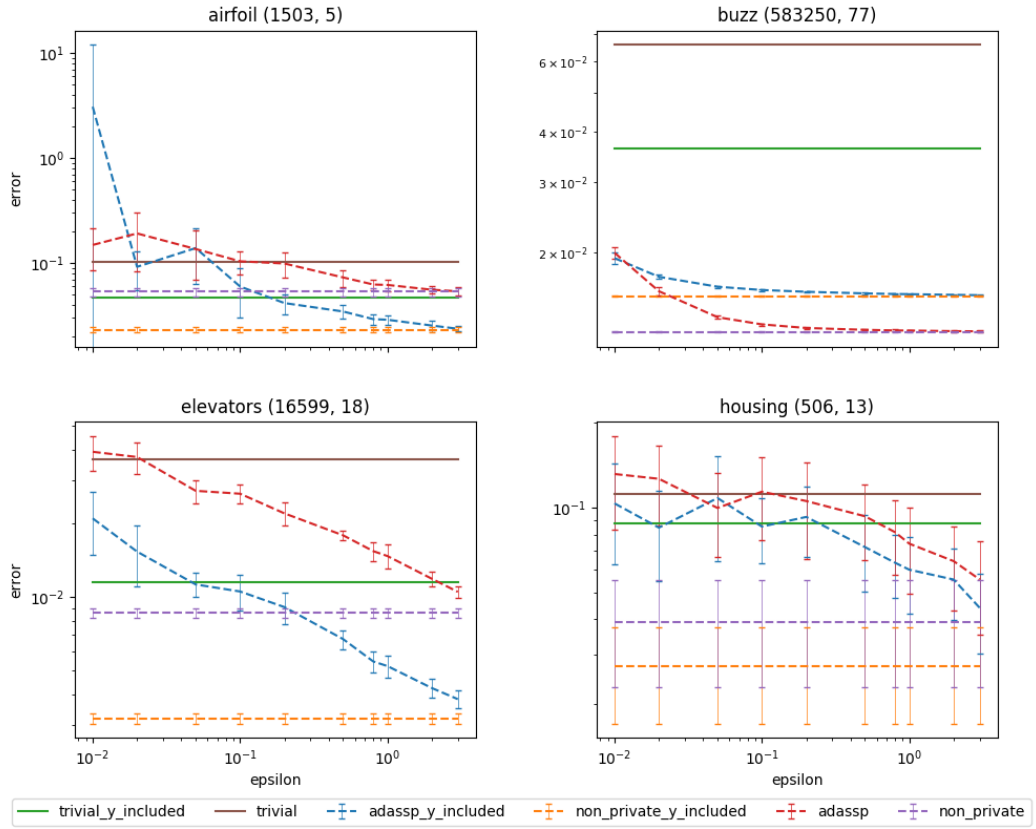

Figure 4.4: Shows differences between two different mapping strategies using row based norms. Postfix "_y_included" is used for graphs with data mapped using row norms also for the response variable $\boldsymbol{y}$ as in Equation 3.4. The convergence of the AdaSSP algorithm towards the non-private solution is clearly visible with both mapping methods.

refer as data analyst, does not have access to the same pre-processed data as the curator, a person whom the original data is trusted and who is in control of the perturbation as in Dwork et al. (2014a). Therefore we change the timing of normalizing and mapping the data to after cross validation step, as we are then able to mimic a situation where the data analyst has different data set to use than the trusted curator.

Wang (2018) normalized all the data including training and test sets together, which mimics a situation where the curator of the data has access to same super pool of data as the data analyst, but these are divided by some third party after some pre-processing protocol. In short, this is a very unlikely scenario, but we can alter the setting a little to make it more reasonable. In Figure 4.5 we compare the setting by Wang (2018)

with a situation where the curator of the data shares from the training data also the column means, column standard deviations, maximum norm of matrix X and maximum norm value from vector $\boldsymbol{y}$. Technically these statistics could be included in the regression coefficient in the case where the statistics can be shared without perturbation as described in Section 3.3.3. If those additional statistics are given to the data analyst by the curator, the results are almost identical. This might not be the case if the curator had chosen not to share these additional statistics from the training data or if he or she had chosen to share the privacy budget over these, which the curator had spent on $X^TX, X^T\boldsymbol{y}$ and $\lambda_{\min}$. Especially when the additional statistics have not been shared, the data analyst with too small sample of data could end up having very noisy versions of these necessary statistics and the predictions could suffer greatly.

4.2.1 Adding the intercept column

As we read through the code of the project Wang (2018) has submitted to github[1], it became clear that intercept column was not used for any of the data sets we have analyzed. Adding the intercept column is such a standard procedure of data pre-processing that is most likely missing just by accident. The normalization of data combined with a mapping to unit sphere inhibits the effect of a missing intercept column, but still the effect should be clear for low dimensional data when we look back to theory in Section 2.1.1.

Again we will pre-process the data after cross-validation step and share the training data parameters for the test set. We use the maximum norm mapping from Equation 3.5 to enforce sensitivity bounds. In Figure 4.6 we compare the prediction error between data with and without an intercept column for all four data sets. It is clear from the graphs that as the number of independent variables grows, the effect of intercept column diminishes for the ADASSP algorithm as expected.

4.2.2 Clipping the data before projection

Robust private linear regression Honkela et al. (2018) projects the data in to smaller subspace of $\mathbb{R}^d$ by clipping the data in to tighter bounds. In Figure 4.7 we use clipping with four different scalars for the σ parameter. We compare the results against data sets without any clipping used prior normalization. Again we also include the trivial solution as a reference. We use clipping as in Equation 3.8, where only upper bound for the data is enforced before pre-processing. For half of the data sets benefits are clear as the epsilon gets larger values. Optimum value for the scalar used in clipping seems to be between 1.3 and 1.6.

[1] (https://github.com/yuxiangw/optimal_dp_linear_regression)

We also try clipping for the test set in Figure 4.8 with the training set clipping parameters. This model has much worse utility and the results were similar for smaller test sets. For Buzz (twitter data set) something quite dramatic happened and the reasons for very different results between the two methods are yet to discover. We are not able exclude the possibility for a simple error in the code as Figure 4.7 shows growing error levels with larger values of ϵ. In any case no clear over all improvement was shown by using this more complex model and we suggest using the *robust private linear regression* as shown by Honkela et al. (2018).

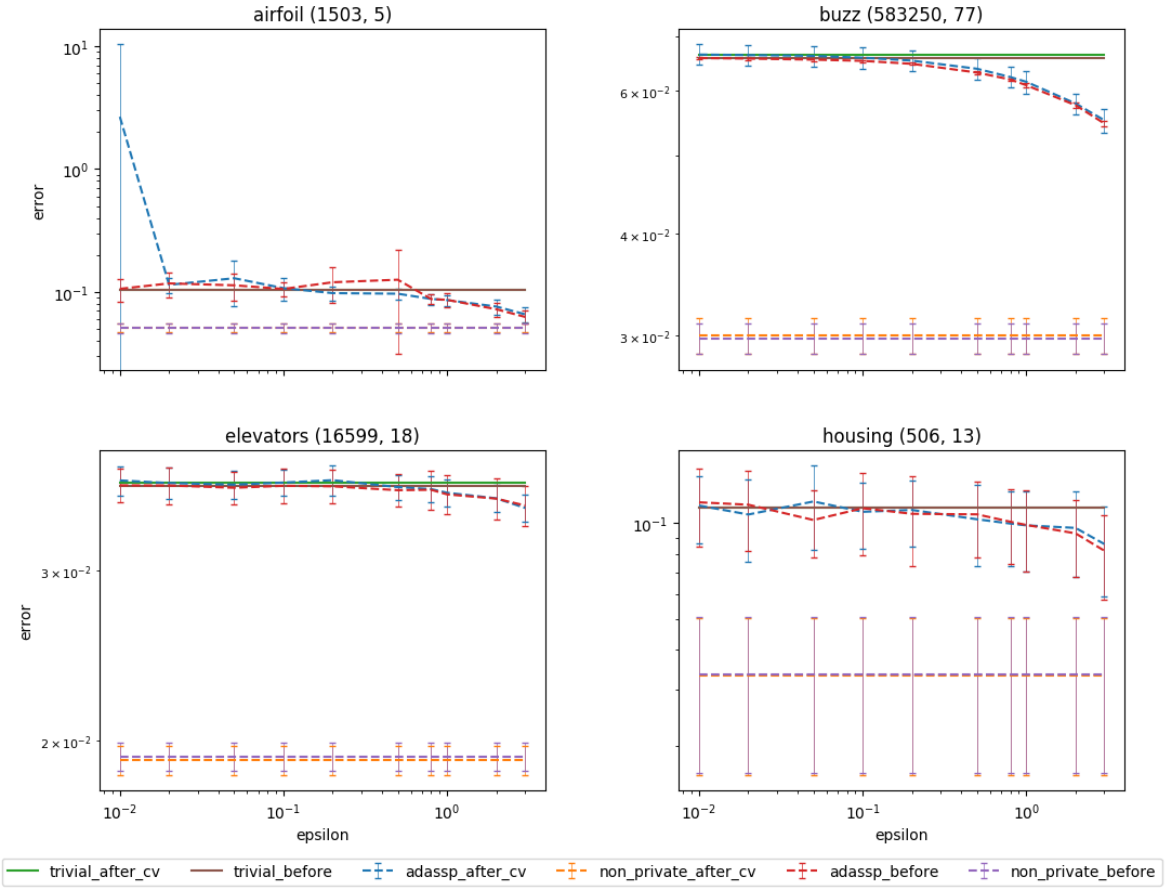

Figure 4.5: The effect of normalization and mapping done to all data before cross-validation split (postfix "_before") compared with normalization and mapping done with the help of training data statistics after cross-validation split (postfix "_after_cv")

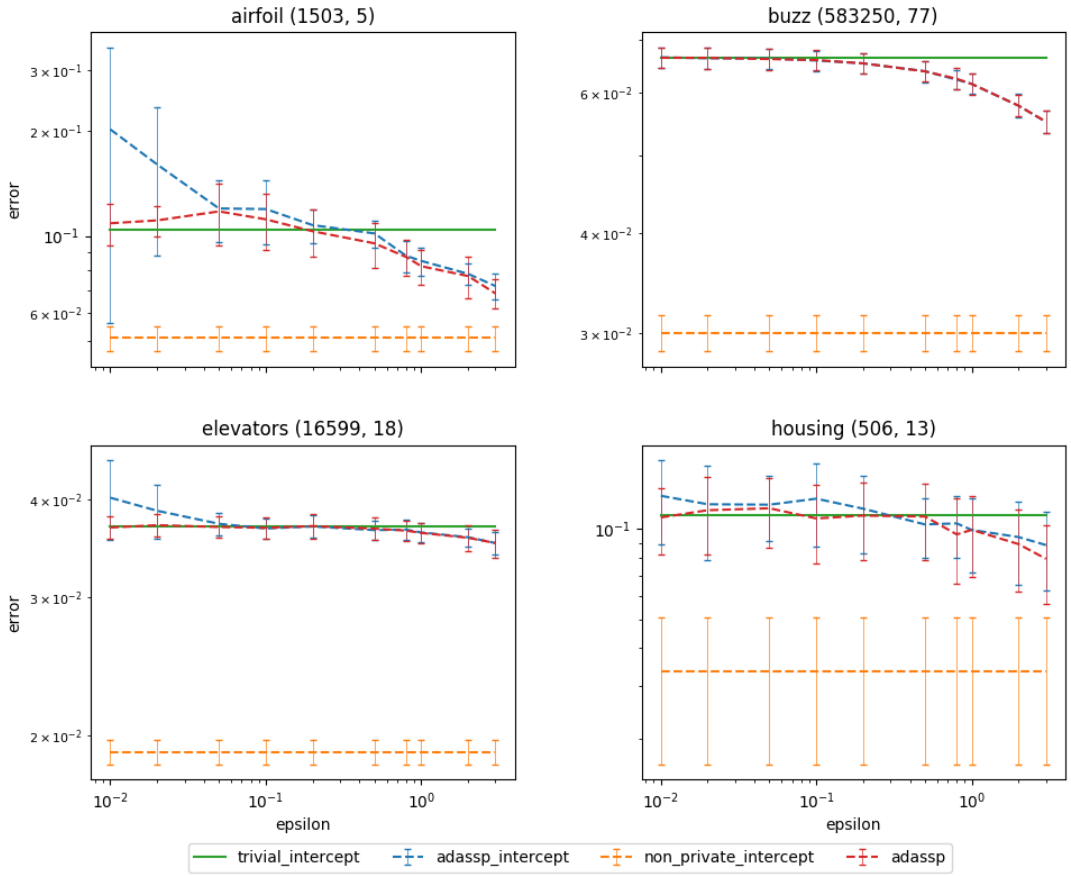

Figure 4.6: Comparison between data with and without intercept column when normalization has been done for the test set with the training set statistics after cross-validation split. Postfix "_intercept" is used when intercept is included.

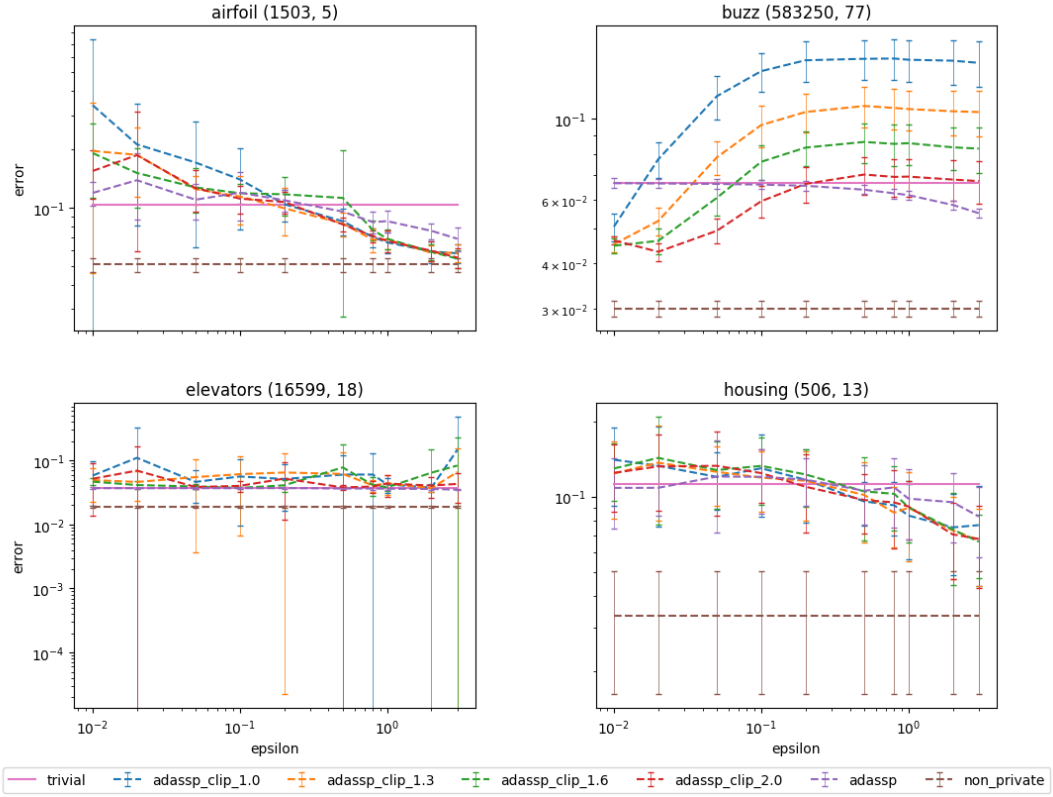

Figure 4.7: Comparison between data with and without clipping before pre-processing. Postfix "_clip_x" is used for the model where clipping was applied to the training data before normalization

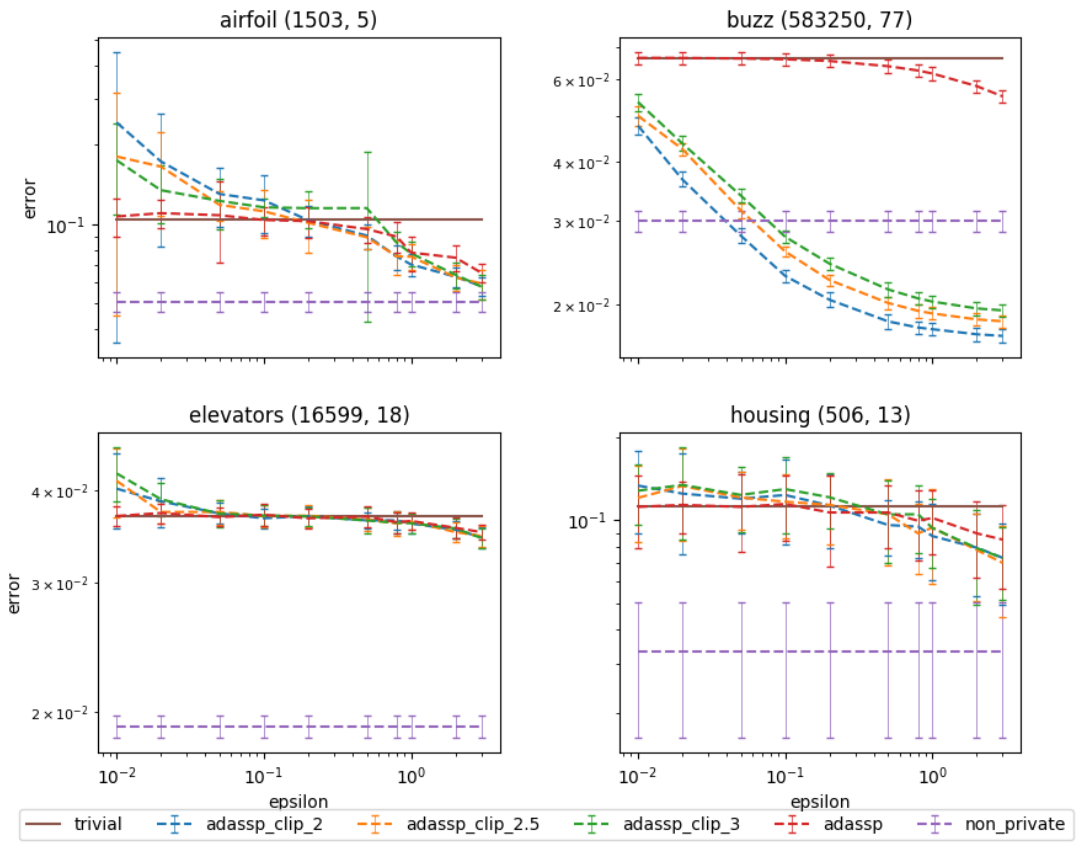

Figure 4.8: Comparison between data with and without clipping before pre-processing, but also the test set is clipped with the bounds calculated from the training set. Postfix "_clip_x" is again used for clipped data.

Chapter 5

Conclusions

We have confirmed that the ADASSP algorithm fulfills the expectations Wang (2018) has set for the algorithm in his paper and is superior to the classic but unstable sufficient statistics perturbation mechanism. The doubts we had regarding the insufficient noise levels for classical Gaussian Mechanism are cleared as the ADASSP is almost as successful also when the scalars for the perturbation parameters are fixed. The unusual pre-processing methods used by Wang (2018) were surprisingly effective in some cases, but the theory of the model was vague. The normalizing decisions were not the reason for the success of the ADASSP algorithm as was shown by using more common standardizing methods in Chapter 4.

The pre-processing decisions were shown to be manifold even in the most simple cases when used in differential privacy. The trusted curator of data set has several options to choose from when releasing the perturbed statistics. None of the methods were able to provide best results every time, but were dependent on the data set in use. As a guideline for the curator we suggest trying *robust private linear regression* by Honkela et al. (2018) alongside with ADASSP algorithm when releasing sufficient statistics. Even though we had weak results by adding the intercept column for the four UCI data sets, we still suggests using it unless shown to lower the estimation accuracy. As a future work for the ADASSP algorithm we leave combining the *Analytical Gaussian mechanism* by Balle and Wang (2018) with the ADASSP algorithm. It will increase the computational strain when running the algorithm, but the results may improve due the smaller levels of noise required for perturbation.

References

Balle, B. and Wang, Y. (2018). Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 403–412. PMLR.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006a). Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006b). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

Dwork, C., Roth, A., et al. (2014a). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

Dwork, C., Talwar, K., Thakurta, A., and Zhang, L. (2014b). Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 11–20.

Galton, F. (1886). Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246–263.

Honkela, A., Das, M., Nieminen, A., Dikmen, O., and Kaski, S. (2018). Efficient differentially private learning improves drug sensitivity prediction. *Biology direct*, 13(1):1.

Kirsch, A. (2011). *An introduction to the mathematical theory of inverse problems*. Applied mathematical sciences (Springer-Verlag New York Inc.). Springer, New York, 2nd ed edition.

Rao, C. R., Rao, C. R., Statistiker, M., Rao, C. R., and Rao, C. R. (1973). *Linear statistical inference and its applications*, volume 2. Wiley New York.

Vu, D. and Slavkovic, A. (2009). Differential privacy for clinical trial data: Preliminary evaluations. In *2009 IEEE International Conference on Data Mining Workshops*, pages 138–143. IEEE.

Wang, Y. (2018). Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain. In Globerson, A. and Silva, R., editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 93–103. AUAI Press.

Weisberg, S. (2005). *Applied linear regression*, volume 528. John Wiley & Sons.

Yan, Xin, k. (2009). *Linear regression analysis : theory and computing*. World Scientific Pub. Co, Singapore ; Hackensack, N.J.

Zhao, J., Wang, T., Bai, T., Lam, K.-Y., Ren, X., Yang, X., Shi, S., Liu, Y., and Yu, H. (2019). Reviewing and improving the gaussian mechanism for differential privacy. *arXiv preprint arXiv:1911.12060*.