Differentially Private Robust Linear Regression

Master's Thesis

Arttu Nieminen

University of Helsinki

10th November 2017

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
Arttu Nieminen

Työn nimi — Arbetets titel — Title

Differentially Private Robust Linear Regression

Oppiaine — Läroämne — Subject
Mathematics

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	November 2017	56 pages

Tiivistelmä — Referat — Abstract

Differential privacy is a mathematically defined concept of data privacy that is based on the idea that a person should not face any additional harm by opting to give their data to a data collector. Data release mechanisms that satisfy the definition are said to be differentially private and they guarantee the privacy of the data on a specified privacy level by utilising carefully designed randomness that sufficiently masks the participation of each individual in the data set. The introduced randomness decreases the accuracy of the data analysis, but this effect can be diminished by clever algorithmic design.

Robust private linear regression algorithm is a differentially private mechanism originally introduced by A. Honkela, M. Das, O. Dikmen, and S. Kaski in 2016. The algorithm is based on projecting the studied data inside known bounds and applying differentially private Laplace mechanism to perturb the sufficient statistics of the Bayesian linear regression model that is then fitted to the data using the privatised statistics.

In this thesis, the idea, definitions and the most important theorems and properties of differential privacy are presented and discussed. The robust private linear regression algorithm is then presented in detail, including improvements that are related to determining and handling the parameters of the mechanism and were developed during my work as a research assistant in the Probabilistic Inference and Computational Biology research group (Department of Computer Science at University of Helsinki and Helsinki Institute for Information Technology) in 2016–2017. The performance of the algorithm is evaluated experimentally on both synthetic and real-life data. The latter data are from the Genomics of Drug Sensitivity in Cancer (GDSC) project and consist of the gene expression data of 985 cancer cell lines and their responses to 265 different anti-cancer drugs. The studied algorithm is applied to the GDSC data with the goal of predicting which cancer cell lines are sensitive to each drug and which are not. The application of a differentially private mechanism to the gene expression data is justifiable because genomic data are identifying and carry highly sensitive information about e.g. an individual's phenotype, health, and risk of various diseases.

The results presented in the thesis show the studied algorithm works as planned and is able to benefit from having more data: in the sense of prediction accuracy, it approaches the non-private version of the same algorithm as the size of the available data set increases. It also reaches considerably better accuracy than the three compared algorithms that are based on different differentially private mechanisms: private linear regression with no projection, output perturbed linear regression, and functional mechanism linear regression.

Avainsanat — Nyckelord — Keywords
Differential privacy, Bayesian linear regression, drug sensitivity prediction, machine learning

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Supervisor: Antti Honkela

Contents

Chapter 1

Introduction

In today's world, large quantities of data are collected and utilised almost everywhere from healthcare and medical research (Naveed et al., 2015) to online entertainment services on the Internet (Narayanan and Shmatikov, 2007). Not only are the amounts of data ever-expanding, the methods of data analytics are constantly improving, resulting in increasingly accurate analysis (Dwork and Roth, 2014, page 213). One example of the rapid progress is the advancement of genomic analysis (Naveed et al., 2015): By studying an individual's DNA, analysts can already tell a lot about e.g. their phenotype, health, risk of various diseases, and longevity. As genomic research progresses, gene technology improves, and the quantity of collected genomic data grows, the amount of information genes can tell us only increases. This raises a question of data privacy: since genomic data carry so much highly personal information, careless handling of the data can place involved individuals in a real danger of having their privacy compromised. The same applies to other types of data, and even rather mundane facts can sometimes reveal more sensitive information (Dwork and Roth, 2014, page 219).

Differential privacy is a practically motivated and mathematically defined concept of data privacy, introduced originally by Dwork et al. (2006). It is based on the principle that a person should not face any additional harm by giving their data out to a data collector — thus, data release mechanisms should not leak sensitive information. Differential privacy is a property of a mechanism that satisfies a certain rigorous mathematical definition (to be presented later), and algorithms that fulfil the property are called differentially private. The idea is that the output of a differentially private mechanism does not reveal if any single individual opted to include their data in the input data set or not because the probability of each possible output is guaranteed to be almost the same in either case. Differentially private mechanisms are often based on adding specifically designed random noise at some step of the algorithm in order to protect sensitive data (Sarwate and Chaudhuri, 2013). The aim of the research on the field of differential privacy is

to develop tools for the purpose of privacy-aware data analysis: mechanisms that allow accurate analysis while also protecting individuals' privacy (Dwork and Roth, 2014, page 215).

Robust private linear regression is a differentially private mechanism designed by A. Honkela, M. Das, O. Dikmen, and S. Kaski (Honkela et al., 2016). During my job as a research assistant in the Probabilistic Inference and Computational Biology research group (Department of Computer Science at University of Helsinki and Helsinki Institute for Information Technology) in 2016–2017, my contribution to the project was to implement a new version of the mechanism and refine it by creating more robust ways to deal with the model parameters. During the project, the algorithm ended up facing even more changes, and the amount of available real-life data used in the experiments also increased. The updated algorithm and the experimental results are presented in the new version of paper (Honkela et al., 2017) and in this thesis. The details of my own contribution are specified in at the end of the thesis in Section 5.2.

This thesis consists of a theoretical and an experimental part. Chapter 2 covers the basics of differential privacy and its theory: first, the idea and some intuition behind the concept are presented, then the mathematical definitions and some of the most important properties are listed. Chapter 3 introduces the robust private linear regression algorithm in detail along with the use case dealing with drug sensitivity prediction, the used data, and the performed tests. The experimental results are presented and discussed in Chapter 4, and the conclusions are further discussed in the last chapter. The main sources are Dwork and Roth's book on differential privacy (Dwork and Roth, 2014) and Honkela et al.'s paper introducing robust private linear regression (Honkela et al., 2017). In order to adequately follow throughout the thesis, the reader should be equipped with basic knowledge about linear algebra, probability calculus, and Bayesian inference, although certain parts of the thesis are possibly comprehensible to readers with little mathematical background.

Acknowledgements

I want to express my warmest thanks to my supervisor Antti Honkela, who has patiently guided me past the many stumbling blocks encountered along the way. The other members of the research group have also been eager to help whenever needed. I learnt a lot during my stay in the group — not only about differential privacy but also about the nature of academic work — and I gained lots of hands-on experience with scientific computing and programming. I also acknowledge the computational resources provided by the Aalto Science-IT project.

Chapter 2

Differential privacy

2.1 Idea

Vast amounts of data are collected by research institutes, government entities, hospitals, and companies (Sarwate and Chaudhuri, 2013) for the purposes of research, development, marketing, and compilation of statistics. Medical research requires data collected from patients in order to study the causes and consequences of illnesses and to develop new treatments (Garnett et al., 2012). Companies use increasingly more client-related data to develop better products, to guide their decision-making, and to design more effective marketing strategies (Schmarzo, 2013, Chapter 1). Government statistics offices collect, store and release vast demographic statistics (Sweeney, 2002) which are needed to make informed and wise decisions concerning the future of the society.

These data typically contain highly sensitive personal information regarding individual people (Sarwate and Chaudhuri, 2013), and a data leakage could potentially risk the privacy of affected individuals, causing severe consequences: Despite being illegal in many countries and states, discrimination still occurs on the basis of e.g. gender, age, religion, ethnicity, sexual orientation and medical status related to e.g. disability (TNS Opinion & Social, 2015b). Employers might fire or refuse to hire people with qualities they find undesirable such as non-heterosexual or transgender identities (Sears and Mallory, 2011), insurance companies may raise the premiums of clients affected by certain medical conditions (Dwork and Roth, 2014, page 215), and credit card issuers can deny applications from individuals who have questionable financial histories (Consumer Financial Protection Bureau, 2017). Another potential threat are criminals who seek ways to misuse leaked private information for their own gain. Even without the possibility of personal harm like the loss of a job or financial stability, many people would still prefer to limit the access to their private data such as their genomic information (Naveed et al., 2015).

According to TNS Opinion & Social (2015a), most people in the European Union

care about the privacy of their data and are concerned about the risks of having their personal information exposed. Therefore, poor privacy policies and the possibility of a data leakage might decrease individuals' eagerness to provide their personal information for various data collectors. As having less data available lowers the accuracy of statistical analysis, a rational aim for analysts is to ensure the security of the data in order to gain individuals' trust. Causing no unnecessary harm for research subjects is also an ethical goal of scientific research (Nuremberg Military Tribunals, 1949). Data analysts considering these issues may thus seek to use methods that guarantee the privacy of data and minimise risks but also allow accurate analysis results.

The concept of differential privacy is one approach to designing such methods and was first defined by Dwork et al. (2006). Dwork (2006) proved that it is impossible to achieve the kind of privacy that would guarantee that the privatised information does not enable learning anything about any individual that cannot be learnt without access to the privatised data: the problem is that there may be arbitrary auxiliary data available from sources other than the data set itself, and that combined with the privatised data can potentially reveal sensitive information. Instead, Dwork (2006) proposed defining privacy with the goal of ensuring that having one's information in a data set should not significantly increase any individual's risk of facing harm. The arising privacy definition is called differential privacy. A more thorough introduction in the topic is given by Dwork and Roth (2014) which will be used as the main source in this chapter.

The setting in differential privacy is illustrated in Figure 2.1, as defined by Dwork and Roth (2014) (Section 2.1): It consists of a population of individuals, data, a data curator, and an outside data analyst or adversary. Individuals provide their data to a data curator who acts as a privacy wall between the private data and the outside world. The curator's job is to protect the data and to decide what information can be given outside so that the released information is useful for a data analyst who wants to make inference about the population, but so that the released information does not reveal any single individual's sensitive data. It is assumed the information which the curator does not release stays private inside the privacy wall and anything the curator releases becomes completely public: on the outside, the data analyst and the adversary have exactly the same access to the released information. This is where the setting differs from for example cryptography, where techniques are based on the assumption that the adversary does not have access to all of the information the analyst has (Dwork, 2006, page 2) — in contrast, in the differential privacy context, the analyst and adversary can be the same person (Dwork and Roth, 2014, page 224). In addition, differential privacy can be accompanied by cryptographic methods inside the privacy wall (Dwork and Roth, 2014, page 231).

Dwork and Roth (2014) (Section 2.1) describe the two types of differential privacy models, offline and online models: In offline models, the curator only publishes information once. The release mechanism has to be carefully designed for each data analysis goal

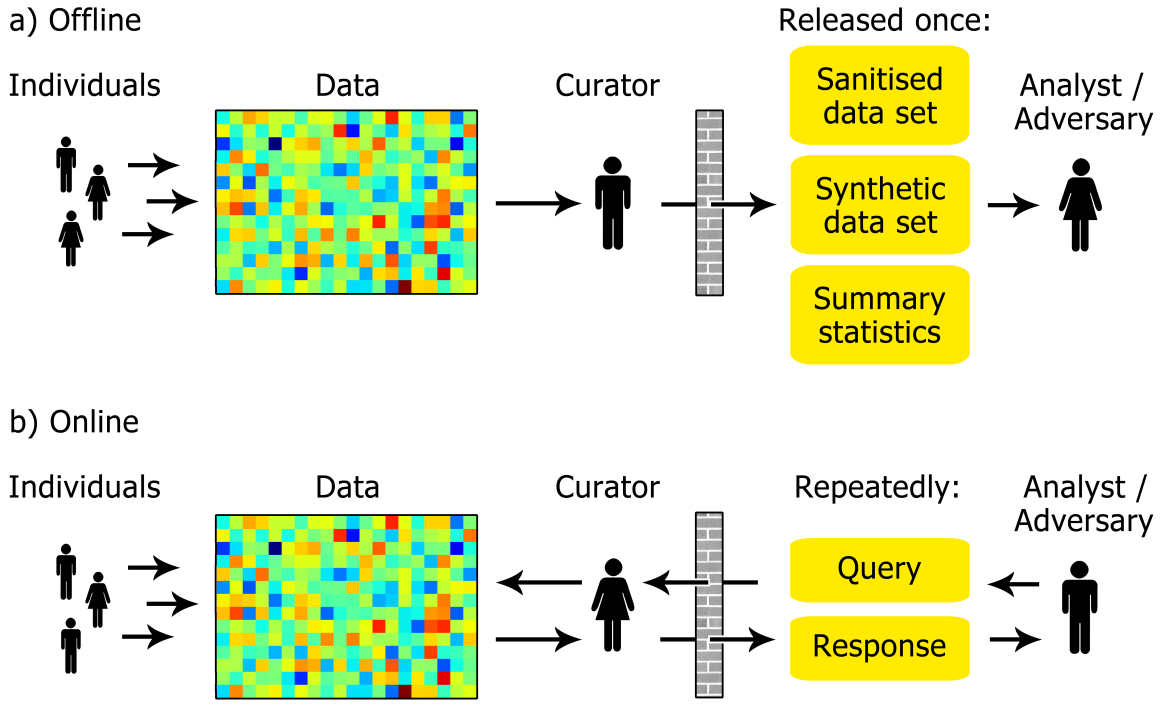

Figure 2.1: **The data release setting in differential privacy.** a) In offline models, the curator decides what information is released, and it is only released once. b) In online models, the analyst (or an adversary) repeatedly makes queries regarding the data, and the curator decides the response to each query. Both the curator and the analyst can adjust their next actions based on the query-response history.

as the data analyst has to be able to make their inferences using only the information the curator has made public. On the other hand, when the goals of the analysis are known in advance, the curator can decide on the optimal release method and adjust the amount of added random noise to be enough for preserving privacy but not too much to destroy accuracy. The curator has many options regarding what kind of information to publish. One alternative is to prepare another data set that is close to the original private data set in any sense the analyst is interested in but does not leak private information. The published data set can be for example a sanitised version where all sensitive information has been removed, or it can be a completely synthetic data set that preserves the interesting structures and patterns in the data. This method has the handy property that the analyst can use the published data in rather similar ways as they would use the actual private data. Another alternative is to publish data summary statistics which in some models are

enough for performing the desired analysis. The other alternative are online models, in which the analyst repeatedly makes queries to the curator who studies the private data set and releases responses. Both the curator and the analyst (or adversary) remember all previous queries and responses and can use them to decide their next query or response. This is a flexible model for analysis settings but it can be a challenging task for the curator to keep track of the query and response history and ensure the next response does not reveal sensitive information when combined with all the previous responses.

The goal of a cleverly designed differential privacy mechanism is to allow the analyst to learn useful information about the studied population but make it nearly impossible for the adversary to learn anything about any single individual (Dwork and Roth, 2014, page 215). This is not a trivial aim, as the data can potentially be highly complex and have delicate patterns that are carried on in a carelessly planned release mechanism. Adversaries can then exploit these loopholes using various attacks to uncover the hidden information. As discussed next, these kind of privacy leaks have been demonstrated by several researchers.

One of the more famous examples of a real-life privacy breach was demonstrated by Narayanan and Shmatikov (2007), using the so called Netflix Prize data set: Netflix, an online video streaming and DVD-rental service, held a competition in 2006–2009 promising an award of one million US dollars to the team who could design the best collaborative filtering algorithm to be used to predict user film ratings and to work as a film recommendation algorithm. The company provided the contestants with a data set comprising film ratings from nearly 500,000 anonymised Netflix users. Narayanan and Shmatikov (2007) used film ratings from the Internet Movie Database (IMDb) as side information and showed their de-anonymisation attack could successfully match some anonymous Netflix users to their public IMDb profiles. The attack was based on the notion that even a relatively small subset of ratings for less popular films can be sufficient to form a rating fingerprint that identifies the user. If an adversary has access to some ratings by an individual, for example via IMDb, public blog posts, or simply a chat with the person in question, the adversary could then use this knowledge to identify the user's entire Netflix ratings in the 'anonymised' data set, possibly gaining access to a great number of ratings not publicly known. In some cases, the private ratings of certain films could be used to draw delicate conclusions about users, such as their political stance or sexual orientation. Indeed, in 2009 four Netflix users filed a lawsuit against the company, among them a lesbian mother who argued the insufficiently anonymised data set could possibly enable hostile parties to reveal her orientation, potentially causing harm to her and her family (Singel, 2009). The lawsuit lead to Netflix cancelling their planned sequel competition that would have released a data set containing even more identifying information (Singel, 2010).

This method of identifying anonymised data records by matching side information to

weakly anonymised data is called a linkage attack (Dwork and Roth, 2014, page 217). It was proven useful also by Sweeney (2002) who was able to identify the personal medical records of Massachusetts governor by using publicly released data: The Group Insurance Commission (GIC) had collected medical data from 135,000 Massachusetts state employees and their families, removed personally identifiable information such as names and exact addresses, and released the data for research use. Sweeney acquired these data and also bought a copy of the local voter registration list. As both data sets included the ZIP codes, genders and birth dates of the individuals, and the voter registration list also contained the names and addresses, she was able to use these attributes to match some individuals between the two data sets. The contemporary Massachusetts governor William Weld had his data in both data sets and also happened to have a unique combination of ZIP code, gender, and the date of birth, which enabled Sweeney to identify the governor's personal medical records in the GIC data set. According to Sweeney (2000), 87% of the United States population have such a combination of these three attributes that likely identifies them. Conclusively, releasing data sets that include these characteristics severely risks the privacy of the related individuals.

For the data curator, anonymising the data set is thus a challenging task, as it is usually impossible to control the available side information and therefore it is difficult to know which data features, many of them seemingly harmless, can be used in a linkage attack. Some identifiable attributes may also be profitable for a data analyst, in which case removing them from the data may be counterproductive. As argued by Dwork and Roth (2014, page 216), in order to perfectly anonymise the data set it needs to be purged so extensively the remaining data are no longer very useful for analysis.

As mentioned earlier, an alternative to releasing an anonymised data set is to only publish certain summary statistics computed from the private data. However, according to Dwork and Roth (2014, page 218), this method as such also poses privacy risks. Consider a simple example data set presented in Table 2.1, containing a number of individuals and their medical status concerning an illness or other medical condition, "1" meaning the person has the condition and "0" meaning the person is not affected by the condition. Now assume the curator decides to only release the number of samples and the average condition status in the data set (the percentage of individuals affected by the condition). An adversary who knows the statuses of all except one individual, for example Mickey, can use this side information to unmask Mickey's medical status by a simple computation: The number of samples times the average gives the sum of statuses S in the private data set. The adversary can then subtract the sum of all known statuses from S, revealing the remaining status and the fact that Mickey has the condition. More generally, if the adversary knows the attribute values $x_1, \ldots, x_{n-1}$ and the released average status $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and the number of samples n, the remaining hidden status x_n can be

computed as

$$n\bar{x} - \sum_{i=1}^{n-1} x_i = \sum_{i=1}^{n} x_i - \sum_{i=1}^{n-1} x_i = x_n.$$

Name	Has condition X
Huey	0
Dewey	1
Louie	1
Daisy	0
Goofy	0
Mickey	1

Table 2.1: An example of a data set containing binary medical information (1 = true, 0 = false).

This is an example of a differencing attack where statistics or queries computed from two versions of a data set are compared in order to make conclusions about the differing part of the data sets (Dwork and Roth, 2014, page 218). It can also prove useful in breaking the privacy of genomic information. Dwork and Roth (2014, page 218) mention single nucleotide polymorphism (SNP) data as an example of sensitive information: SNPs are genetic variations that occur in single nucleotides in DNA. The different forms, alleles, may have different frequencies in population and are used in medical research since a certain allele may indicate a statistical risk of a particular disease (Alwi, 2005). Research data may therefore contain the allele frequencies (aggregate statistics) for many SNPs in the case group and the healthy control group. If an individual's genomic data are available, a differencing attack could in theory be used to determine if the individual is in the case group and has the disease (Dwork and Roth, 2014, page 219). For this reason, researchers consider raw SNP data too risky to be made freely available online (Naveed et al., 2015) and National Institutes of Health and Wellcome Trust have restricted access to their genomic aggregate data (Dwork and Roth, 2014, page 219).

Dwork and Roth (2014, page 218) explain that the data curator could also try to restrict the queries in some manner and only reply to those who are considered safe: the curator could for example only allow queries over larger data sets and refuse to answer to queries that relate to a single individual. In order to protect the data against a differencing attack like the one presented earlier with Table 2.1, the curator would have to be able to identify all possible combinations of queries that would allow the adversary to exploit such an attack. According to Dwork and Roth (2014, page 218), it might not be an easy or computationally feasible task to determine which queries can be answered to without

leaking any protected information, and in some cases the refusal to respond to a specific query can itself prove to be revealing.

As noted by Dwork and Roth (2014, page 219), the curator also cannot safely rely on the presumption that ordinary or unremarkable facts would be safe to be released in all situations: For instance, grocery purchases may reveal information about an individual's medical conditions such as allergies, coeliac disease or diabetes. Since data can contain highly complex dependencies, correlations and patterns that link such ordinary facts and more sensitive information, it is not easy to know which facts could be considered safe to be published. According to Dwork and Roth (2014, page 219), sometimes the curator might also be content with providing protection for the typical members of the data set while potentially risking the privacy of a few less typical individuals. However, these kind of mechanisms are potentially ethically questionable, and the non-typical individuals might be exactly those who are the most susceptible to potential harm caused by privacy breaches.

Differential privacy is designed in such a way that it provides protection from all of the aforementioned issues (Dwork and Roth, 2014, page 216). As explained by Dwork and Roth (2014, page 230), its starts from the idea that when an individual considers whether to consent to giving their data to a curator, they weigh the risks and assess if the consent would likely lead to facing negative consequences that could be avoided by refusing to hand over their data. If the curator can guarantee that consenting likely will not cause any additional harm, a rational individual will then be convinced to giving their data to the curator for a small reward or for other personal or altruistic reasons. This idea is formulated mathematically in the definition of differential privacy, which is a property of a privacy-preserving data release mechanism and not an algorithm as itself. Differentially private mechanisms utilise carefully designed randomness in their logic, and can be mathematically proven to guarantee that it is very likely the probability of getting each possible output is almost the same whether any single individual opts in or out of the data set (Dwork and Roth, 2014, page 216). Consequently, given the released information, the adversary party cannot determine if any specific individual is included in the data set or not since the probability of outputting the observed information is very similar in either case and the observed output thus provides no strong evidence in favour of either option. Since the adversary cannot be sure of any individual's participation, they cannot make any inference about any single person. Therefore, the individuals' sensitive data are protected.

Differential privacy is not based on keeping the data release mechanism private — on the contrary, the mechanism itself can be made public since privacy arises from the involved randomness instead. Differentially private mechanisms are based on utilising protective randomness (Dwork and Roth, 2014, page 225): they typically add random noise at some step of the algorithm or make random selections. As explained by Dwork

and Roth (2014, page 226), randomness is necessary, since deterministic mechanisms always produce the same output with the same input data set and are thus vulnerable to differencing attacks: Trivially, data privacy can be protected by simply always outputting some nonsense data that do not carry any useful information, but for a non-trivial algorithm, there exist such a query and a pair of data sets differing only at one data point that produce different responses to the query. If an adversary tries out different combinations by changing the input data set one point at a time until they find such a query and a pair of data sets, and if they know the actual real data set is one of these two options, they could then determine the values of the differing data point.

On the other hand, the added randomness necessarily makes the results of the analysis more uncertain. However, since the guarantee of differential privacy potentially convinces more people to give their data, the increased number of samples may quite possibly overcome the experienced loss of accuracy in data analysis. The challenging task is to inject just the right amount of randomness that preserves both the privacy and the interesting patterns of the data (Dwork and Roth, 2014, page 216).

As will be seen, the definition of differential privacy also enables the measurement of the privacy level of any mechanism that satisfies the definition, and the measurement of the amount of privacy loss incurred by observing a certain output. This makes it possible to compare different differentially private mechanisms against each other. The definition is very general and thus applicable to any data sets and data analysis models. The theory of differential privacy elegantly builds on itself and provides many useful theorems for designing privacy-preserving algorithms. The basic definition and the most relevant properties are presented in the following two sections.

2.2 Definitions

The basic definitions and the theory of differential privacy are presented here as by Dwork and Roth (2014, Section 2.3), except that instead of the histogram notation, data sets are treated as multisets consisting of elements that represent individual data points (often rows of a data matrix). Let U denote the universe of all possible data elements. A multiset is a mapping that maps each element in the universe U to a natural number representing the number of instances of the element in the multiset.

Definition 2.1. *(Multiset)* Given a universe U, a multiset is a mapping

$$m : U \to \mathbb{N} = \{0, 1, 2, \dots\}.$$

The size of the multiset is the number of its elements, with each instance counted in.

Definition 2.2. *(Size of a multiset)* The size of the multiset $m : U \to \mathbb{N}$ is

$$|m| = \sum_{u \in U} m(u).$$

Let $\mathcal{D}$ denote the set of all possible multisets of universe U. The distance between multisets is the number of differing instances between them.

Definition 2.3. *(Distance between multisets)* Distance between multisets $x \in \mathcal{D}$ and $y \in \mathcal{D}$ is

$$\text{dist}(x, y) = \sum_{u \in U} |x(u) - y(u)|.$$

In this thesis, multisets x and y are called neighbours if $\text{dist}(x, y) \leq 2$ and $|x| = |y|$, that is, either they are the same multiset or one of the multisets can be transformed into the other by changing one element into another one. Note that in each application, the concept of neighbours has to be considered carefully (Dwork and Roth, 2014, page 233). In this work, it is appropriate to use the multiset notation and define neighbours as multisets of the same size who differ at most one instance, but in some other context, a different definition may be needed. For example, social networks are more sensibly presented as graphs consisting of nodes and edges between them. Defining neighbouring graphs based on differing edges or differing nodes lead to very different concepts of differential privacy, one of which might be too strong and thus unsuitable for a certain application (Dwork and Roth, 2014, page 234).

In order to mathematically define differential privacy, a few more definitions are needed. A probability simplex over a discrete set consists of all possible discrete probability distributions over the set (Dwork and Roth, 2014, Definition 2.1).

Definition 2.4. *(Probability simplex)* Let A be a discrete set. The probability simplex over set A is

$$\Delta(A) = \left\{ p \in \mathbb{R}^{|A|} : p_i \geq 0 \text{ for all } i = 1, \ldots, |A|, \sum_{i=1}^{|A|} p_i = 1 \right\}.$$

A randomised algorithm is a non-deterministic algorithm which applies randomness in its inner workings. Given an input, the output of a randomised algorithm is not determined — instead, a probability distribution over all possible outputs is defined (Dwork and Roth, 2014, Definition 2.2).

Definition 2.5. *(Randomised algorithm)* A randomised algorithm $\mathcal{M}$ with domain $\mathcal{D}$ and discrete range $\mathcal{R}$ (also denoted by $\mathcal{M} : \mathcal{D} \to \mathcal{R}$) is an algorithm associated with a mapping $M : \mathcal{D} \to \Delta(\mathcal{R})$. On input $x \in \mathcal{D}$, algorithm $\mathcal{M}$ outputs $\mathcal{M}(x) = r$ with probability $(M(x))_r$ for each $r \in \mathcal{R}$. The probability space is over the random choices of the algorithm $\mathcal{M}$.

Using these definitions, the concept of differential privacy can now be defined (Dwork and Roth, 2014, Definition 2.4).

Definition 2.6. *(Bounded differential privacy)* A randomised algorithm $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ is (ϵ, δ)-differentially private with $\epsilon > 0$ and $\delta \geq 0$, if for all subsets $S \subseteq \mathcal{R}$ and for all data sets $x, y \in \mathcal{D}$ such that $\text{dist}(x, y) \leq 2$ and $|x| = |y|$,

$$P(\mathcal{M}(x) \in S) \leq e^{\epsilon} P(\mathcal{M}(y) \in S) + \delta,$$

where the probability space is over the random choices of algorithm $\mathcal{M}$. If algorithm $\mathcal{M}$ is $(\epsilon, 0)$-differentially private, it is said to be ϵ-differentially private. The parameter $\epsilon > 0$ is called the privacy parameter or the privacy budget.

In this thesis, the focus is on the stricter special case where $\delta = 0$. For a sufficiently small value of ϵ, an algorithm fulfilling the ϵ-differential privacy definition protects the privacy of any individual data point in the input space: Given two neighbouring data sets differing at most one element, the probability of getting any certain output from the algorithm is nearly the same no matter which of the two data sets is given as the input. For the adversary who observes the output of the algorithm, this means it is practically impossible to infer which data set was given as the input since any two neighbouring data sets would have produced the output with almost equal probabilities. Conversely, if the differential privacy condition does not hold and instead some data set x leads to a certain output with a much larger probability than its neighbouring data set y, the adversary could deduce it is much more likely the input is x rather than y.

If the adversary cannot distinguish between neighbouring input data sets, the individual data points are safe: Since any input data point can be exchanged for another one without significantly affecting the output results, the adversary cannot know if a certain individual person had their data in the input or not. The privacy of the person is thus guaranteed as the adversary cannot reveal the presence or content of their data. Since adversary parties will not find out if someone participates in a medical study carried out in a differentially private manner, individuals can safely give their data to the researches without having to worry that their medical status could be exposed because of this action.

The definition requires the condition holds for all possible outputs and all existing neighbouring data sets. Therefore, it impartially guarantees the privacy of every individual simultaneously and regardless of their qualities.

The privacy parameter ϵ determines how close to each other the output probabilities have to be. Therefore, it denotes the level of privacy: small values of ϵ ensure very strict privacy guarantees and larger values looser ones. Looking at Definition 2.6 (and assuming $\delta = 0$ for now), for a small ϵ close to 0, the term e^ϵ is close to 1 and hence the output probabilities $P(\mathcal{M}(x) \in S)$ and $P(\mathcal{M}(y) \in S)$ have to be almost equal to each other (strict privacy). As the value of ϵ increases, the term e^ϵ gets larger as well and the output probabilities are allowed to be further away from each other (loose privacy). The natural exponential function e^ϵ is strictly increasing so increasing the privacy parameter ϵ increases the privacy level and vice versa. Choosing a suitable value of ϵ for each application is not necessarily a trivial task (Sarwate and Chaudhuri, 2013) although some heuristics have been proposed (Dwork and Smith, 2009).

(ϵ, δ)-differential privacy with $\delta > 0$ is a weaker guarantee than the case with $\delta = 0$. The parameter δ determines how probable it is that two neighbouring data sets produce a certain output with probabilities that are further away from each other than the privacy parameter ϵ alone would allow. For $\delta = 0$, no deviations from the condition are allowed: on every run of the algorithm, for every neighbouring pair of data sets, and for every output, the output probabilities have to be close to each other. For larger $\delta > 0$, given two neighbouring data sets, it is highly likely the probabilities of producing the observed output are sufficiently close to each other as defined by the parameter ϵ — but on the other hand, given an output s, there may exist some neighbouring data sets that have very different probabilities for outputting s (Dwork and Roth, 2014, page 228). The parameter δ therefore defines the allowed probability of a possible privacy breach. In some applications, fulfilling $(\epsilon, 0)$-differential privacy may be a too strong requirement and the relaxation with $\delta > 0$ is reasonable. For any sensible privacy mechanism, the value of the parameter δ should be very small. As noted by Dwork and Roth (2014, page 228), it usually should be smaller than the inverse of any polynomial in the size of the data set.

The Definition (2.6) of differential privacy naturally leads to a way to measure how much privacy is lost when a certain output is observed (Dwork and Roth, 2014, page 228).

Definition 2.7. *(Privacy loss)* For mechanism $\mathcal{M}$ and input data sets $x, y \in \mathcal{D}$, the privacy loss incurred by observing output $r \in \mathcal{R}$ is

$$\mathcal{L}^{(r)}_{\mathcal{M}(x)||\mathcal{M}(y)} = \ln \left[\frac{P(\mathcal{M}(x) = r)}{P(\mathcal{M}(y) = r)} \right].$$

If the output r is more likely to occur with input data set x than y, the privacy loss is positive, and if vice versa, the privacy loss is negative. The parameter ϵ determines how close to zero the privacy loss has to be with neighbouring data sets x and y, and the parameter δ controls how likely it is the aforementioned condition is allowed to be

broken. The definition enables comparing various differentially private mechanisms and their privacy guarantees.

As explained by Kifer and Machanavajjhala (2011), Definition 2.6 is for the so-called bounded differential privacy which assumes the neighbouring data sets are of the same size and differ at most one element. Alternatively, one can define unbounded differential privacy, the only difference between the two options being that the unbounded definition allows the sizes of the neighbouring data sets to differ by at most one. Two data sets are thus called neighbours if one can be transformed into the other by either removing or adding one element.

Definition 2.8. *(Unbounded differential privacy)* A randomised algorithm $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ is (ϵ, δ)-differentially private with $\epsilon > 0$ and $\delta \geq 0$, if for all subsets $S \subseteq \mathcal{R}$ and for all data sets $x, y \in \mathcal{D}$ such that $\text{dist}(x, y) \leq 1$,

$$P(\mathcal{M}(x) \in S) \leq e^{\epsilon} P(\mathcal{M}(y) \in S) + \delta,$$

where the probability space is over the random choices of algorithm $\mathcal{M}$.

Throughout this thesis, the bounded version of differential privacy (Definition 2.6) is used. In that context, it is clear the size of the data set is not a private variable. The two alternatives are not straightforwardly interchangeable and it is important to note which version is used. The basic theory of differential privacy works in a very similar manner with either definition.

2.3 Properties

The theory of differential privacy builds elegantly on top of the basic definitions. This section lists some of the most important and useful properties of differentially private mechanisms. Moreover, some fundamental restrictions are discussed. The presentation of the theorems and proofs mostly follows Dwork and Roth (2014, Sections 2.3 and 3.5) although the previously introduced multiset notation is used instead of the histogram notation.

A crucial property of differential privacy is that it is immune to post-processing: the output of a differentially private mechanism cannot be made less private by any kind of computing or other processing that is performed to the output after its release (Dwork and Roth, 2014, Proposition 2.1). This means adversary parties are not able to break the guaranteed privacy level no matter how much effort they spend on studying the output.

Theorem 2.9. *(Post-processing) Let $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ be an (ϵ, δ)-differentially private mechanism and let $f : \mathcal{R} \to \mathcal{R}'$ be a randomised mapping. Then $f \circ \mathcal{M} : \mathcal{D} \to \mathcal{R}'$ is (ϵ, δ)-differentially private.*

Proof. We assume $f : \mathcal{R} \to \mathcal{R}'$ is a deterministic mapping and prove that post-processing with f preserves differential privacy. The result then follows because any randomised mapping can be expressed as a convex combination of a set of deterministic mappings, and a convex combination of (ϵ, δ)-differentially private mechanisms is (ϵ, δ)-differentially private.

Let $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ be an (ϵ, δ)-differentially private mechanism, let $x, y \in \mathcal{D}$ be neighbouring data sets with $\text{dist}(x, y) \leq 2$ and $|x| = |y|$, and let $S \subseteq \mathcal{R}'$. Denote the preimage of S by $f^{-1}(S) = \{r \in \mathcal{R}, f(r) \in S\}$. Now

$$P(f(\mathcal{M}(x)) \in S) = P(\mathcal{M}(x) \in T) \leq e^{\epsilon} P(\mathcal{M}(y) \in T) + \delta = e^{\epsilon} P(f(\mathcal{M}(y) \in S) + \delta.$$

□

Differential privacy is usually discussed on the level of individuals as the neighbouring data sets are often defined as differing at most one data point. Mechanisms that fulfil the basic Definition 2.6 thus guarantee their results to stay almost the same if any one data entry is changed. However, in some applications data sets may contain clearly distinguishable groups with correlations between the qualities of the group members. For instance, some illnesses and other medical conditions are genetically inheritable. Additionally, families and other groups may share lifestyles and environmental factors which also affect the prevailing properties in the group. While a differentially private mechanism defined as in Definition 2.6 masks the data of any one individual, it does not protect the privacy of bigger groups in the same level as individuals: the adversary might more easily be able to reveal the existence of a group in the data set. Accompanied with useful side information, they might then be able to find out the properties of some individual. For instance, the condition X in the simple example Table 2.1 may be known to be strongly genetically inheritable. If the differentially private mechanism protects the individual data entries on a sufficient privacy level, the adversary cannot find out the medical status of one person, for example Louie. However, the mechanism might not be able to properly mask the existence of a larger group such as a whole family. By studying the correlations between different features (in addition to 'has condition X') in the data set, the adversary might thus be able to find out there is a distinguishable group of three individuals and two of them have the condition. Now assuming it is public knowledge that Huey, Dewey and Louie are brothers, using their public attributes, the adversary might find out the brothers match the revealed group of three individuals in the data set. The adversary could then conclude that Louie has a very high chance of having condition X — a deduction that could not have been made if the privacy mechanism had sufficiently protected the privacy of larger groups.

From Definition 2.6 it straightforwardly follows that ϵ-differential privacy does not instantly get destroyed for larger groups but instead degrades linearly as the size of the

group increases: an algorithm guaranteeing $(\epsilon, 0)$-differential privacy for individuals offers $(k\epsilon, 0)$-differential privacy for groups of size k (Dwork and Roth, 2014, Theorem 2.2). As noted by Dwork and Roth (2014, page 230), the decrease of the privacy level is necessary: if more and more input data points are replaced with different ones, the results of any sensible mechanism should change or they would not really teach the analyst anything useful about the data set.

Theorem 2.10. *(Group privacy) Let $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ be an $(\epsilon, 0)$-differentially private mechanism. Then $\mathcal{M}$ is $(k\epsilon, 0)$-differentially private for groups of size $k \in \mathbb{N}$. That is, for all data sets $x, y \in \mathcal{D}$ with dist$(x, y) \leq 2k$, $|x| = |y|$, and all subsets $S \subseteq \mathbb{R}$,*

$$P(\mathcal{M}(x) \in S) \leq e^{k\epsilon} P(\mathcal{M}(y) \in S).$$

Proof. Let $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ be an $(\epsilon, 0)$-differentially private mechanism, and let $x, y \in \mathcal{D}$ be data sets with $\text{dist}(x, y) \leq 2k$ (data sets x and y differ at most k records). There then exists a chain of $k - 1$ neighbouring data sets $z_1, \ldots, z_{k-1}$ such that $\text{dist}(x, z_1) \leq 2, \text{dist}(z_1, z_2) \leq 2, \ldots, \text{dist}(z_{k-1}, y) \leq 2$. Let $S \subseteq \mathbb{R}$. Now

$$P(\mathcal{M}(x) \in S) \leq e^{\epsilon} P(\mathcal{M}(z_1) \in S) \leq e^{\epsilon} e^{\epsilon} P(\mathcal{M}(z_2) \in S) \leq \cdots \leq e^{k\epsilon} P(\mathcal{M}(y) \in S).$$

□

The corresponding result for (ϵ, δ)-differential privacy warrants $(k\epsilon, ke^{(k-1)\epsilon}\delta)$-differential privacy for groups of size k (Dwork and Roth, 2014, page 230) but it will not be discussed here in more detail.

In many applications, the privacy mechanism may consist of several independent components, or the same statistic may be outputted multiple times during the execution of the algorithm. Therefore, it is essential to understand how strong the combination of differentially private parts is. As explained by Dwork and Roth (2014, page 252), the privacy level of the composition is inevitably lower than the privacy level of its parts: having several different statistics potentially carries more information than one, and if the same statistic is computed and perturbed with noise several times, the average of the perturbed statistics may reveal more about the input than a single perturbed statistic. Aptly, the privacy parameter ϵ can be thought of as privacy budget that is spent with each release of statistics. The composition theorem of differential privacy (Dwork and Roth, 2014, Theorem 3.14) states that the privacy budget of the composition is the sum of the component privacy budgets:

Theorem 2.11. *(Composition) Let $\mathcal{M}_1 : \mathcal{D} \to \mathcal{R}_1$ be an $(\epsilon_1, 0)$-differentially private mechanism and $\mathcal{M}_2 : \mathcal{D} \to \mathcal{R}_2$ be an $(\epsilon_2, 0)$-differentially private mechanism, and let the two mechanisms operate independently from each other. Then mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}_1 \times \mathcal{R}_2$, defined by $\mathcal{M}(x) = (\mathcal{M}_1(x), \mathcal{M}_2(x))$ for all $x \in \mathcal{D}$, is $(\epsilon_1 + \epsilon_2, 0)$-differentially private.*

Proof. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be independent differentially private mechanisms and their composition $\mathcal{M}$ be defined as in the proposition. Let $x, y \in \mathcal{D}$ be two neighbouring data sets with $\text{dist}(x, y) \leq 2$ and $|x| = |y|$, and let $(r_1, r_2) \in \mathbb{R}_1 \times \mathbb{R}_2$ be any output. Now

$$\begin{aligned}
\frac{P(\mathcal{M}(x) = (r_1, r_2))}{P(\mathcal{M}(y) = (r_1, r_2))} &= \frac{P((\mathcal{M}_1(x), \mathcal{M}_2(x)) = (r_1, r_2))}{P((\mathcal{M}_1(y), \mathcal{M}_2(y)) = (r_1, r_2))} \\
&= \frac{P(\mathcal{M}_1(x) = r_1)P(\mathcal{M}_2(x) = r_2)}{P(\mathcal{M}_1(y) = r_1)P(\mathcal{M}_2(y) = r_2)} \\
&= \frac{P(\mathcal{M}_1(x) = r_1)}{P(\mathcal{M}_1(y) = r_1)} \cdot \frac{P(\mathcal{M}_2(x) = r_2)}{P(\mathcal{M}_2(y) = r_2)} \\
&\leq e_1^{\epsilon} e_2^{\epsilon} = e^{\epsilon_1 + \epsilon_2}.
\end{aligned}$$

□

Applying this repeatedly k times leads to the corollary (Dwork and Roth, 2014, Corollary 3.15):

Corollary 2.12. *Let $\mathcal{M}_i : \mathcal{D} \to \mathcal{R}_i$ be an $(\epsilon_i, 0)$-differentially private mechanism for $i = 1, \ldots, k$. Then mechanism $\mathcal{M} : \mathcal{D} \to \prod_{i=1}^{k} \mathcal{R}_i$, defined by $\mathcal{M}(x) = (\mathcal{M}_1(x), \ldots, \mathcal{M}_k(x))$ for all $x \in \mathcal{D}$, is $(\sum_{i=1}^{k} \epsilon_i, 0)$-differentially private.*

A similar result holds for the more general (ϵ, δ)-differential privacy: the composition of (ϵ_i, δ_i)-differentially private mechanism, $i = 1, \ldots, k$, is $(\sum_{i=1}^{k} \epsilon_i, \sum_{i=1}^{k} \delta_i)$-differentially private (Dwork and Roth, 2014, Theorem 3.16). The proof can be found in the book Dwork and Roth (2014, Appendix B). Another more sophisticated result, the advanced composition theorem (Dwork and Roth, 2014, see Theorem 3.20 in Section 3.5.2) enables evaluating the privacy level of differentially private systems that may for example handle situations where an individual's data are scattered in several different data sets that can be used independently by differentially private mechanisms.

When designing and using differentially private systems, it is essential to be aware of its restrictions and understand what exactly is protected by differential privacy and what is not. In every application, it is crucial to carefully evaluate the assumptions about the data, consider what exactly needs to be protected and what not, think about the utility goal and the required level of privacy, and mathematically prove that the differential privacy condition is satisfied. These remarks are noted by Dwork and Roth (2014) and Kifer and Machanavajjhala (2011), and they are briefly discussed below.

The basic idea of differential privacy is to guarantee that individuals will not face any additional harm from opting to have their data in the data set, but this does not, however, warrant protection against any arbitrary harm. As an example, Dwork and Roth (2014, page 215) mention how a medical study may reveal that people who smoke cigarettes have

an increased risk of getting cancer. Based on this information, health insurance companies might raise the premiums of smokers. In such a situation, a person who regularly smokes is thus harmed by the study, but this does not violate the differential privacy condition: If the research was carried out in a differentially private manner, the outcome does not reveal the smoking habits of any single person who took part in the study, and instead, the insurance company would have to get the information about the person's smoking habit from somewhere else. According to the definition of differential privacy, the results of the study would be the same whether the person takes part in the study or not, so the experienced harm is not something additional due to being in the data set.

The basic definition of differential privacy also only guarantees that the output would likely be similar whether any certain data point is in the input or not. The idea is that this prevents the adversary from finding out the existence of the data point in the data set since it could be switched out without affecting the output results. However, in some contexts, switching out a data point does not always remove all evidence of its existence. If all entries are independent, all evidence vanishes along with the entry itself, but in other situations, the data points may have complex dependencies between them. In social networks, one person can introduce people to each other, creating links that would not have formed otherwise. In such a situation, just removing the person from the data set does not hide all evidence of the existence of the said person since the links between other people remain. This problem is addressed by Kifer and Machanavajjhala (2011) who discuss and analyse the restrictions of differential privacy. One of their main points is that it is impossible to build useful and functional differentially private mechanisms without making any assumptions about the data. Therefore, possible links and patterns in the data have to be carefully considered.

Differential privacy also does not create privacy if none exists in the first place: if everything is already revealed, it cannot be made private again. Furthermore, as pointed out by Kifer and Machanavajjhala (2011), the theory of differential privacy is compatible with itself but not always with other privacy mechanisms. In particular, sometimes the accuracy of the results is so important it is necessary to release certain exact statistics computed from the data. If the same data are later used to release other, differentially private statistics, careful consideration has to be taken in order to prevent additional information leakages due to the combination of perturbed and exact statistics. The problem is that while differentially private statistics protect individual data entries, they may reveal correlations in the data that together with the exact statistics expose the original data. This problem is illustrated with examples by Kifer and Machanavajjhala (2011) who also propose how it could be dealt with.

In differential privacy settings, there is always a trade-off to be made between privacy and utility (Sarwate and Chaudhuri, 2013): increasing the level of privacy decreases the accuracy of the results and vice versa. A mechanism can be designed to meet extremely

high privacy requirements but it is completely useless if the results are so inaccurate they are not of any use to the analysts.

As stated by Dwork and Roth (2014, pages 211 and 215), all privacy mechanisms have fundamental restrictions, and the goal is to postpone bumping into these limitations as long as possible by striving for clever design. Despite the limitations, the definition of differential privacy is usually very strict, and in many situations, less than that would still be sufficient. Particularly, all small values of ϵ guarantee approximately the same level of privacy, and failing to satisfy the differential privacy condition with a certain small value of ϵ does not necessarily mean the mechanism poses severe privacy risks (Dwork and Roth, 2014, page 234). Moreover, the definition of $(\epsilon, 0)$-differential privacy guarantees the privacy of every possible data set in the space, which is a very strong requirement since some data sets may be extremely rare in practice (Dwork and Roth, 2014, page 235). Furthermore, differential privacy also protects against theoretical adversaries with unlimited computational power or arbitrary auxiliary information (Dwork and Roth, 2014, pages 232–233). Carefully built differentially private mechanisms can thus work as powerful tools for data curators and analysts.

2.4 Differentially private mechanisms

2.4.1 Different approaches

As already explained, differential privacy requires that a certain amount of randomness is applied in the privacy mechanism. This can be done in various ways, usually by adding random noise at some stage of the algorithm. Sarwate and Chaudhuri (2013) describe four key approaches for building differentially private algorithms: input perturbation, output perturbation, objective perturbation, and exponential mechanism. Input perturbation, as the name suggests, guarantees differential privacy by adding random noise directly to the data itself. Output perturbation injects noise into the output of the algorithm and instead releases the perturbed statistics. Objective perturbation involves adding noise to the objective function of the algorithm, ensuring that the intermediate results are private: as Theorem 2.9 states, any further computation still preserves the privacy. Another very general method is exponential mechanism introduced by McSherry and Talwar (2007): it enables selecting the output from several possible choices based on their quality in a way that both satisfies differential privacy and makes the high-utility outputs exponentially more likely to be picked. The choice of the suitable method and the details depend on the data, the release setting, and the intended use of the released information. Straightforwardly replacing the non-private steps of an algorithm with privacy-preserving ones does not guarantee the optimal method — instead, designing the method with privacy

as a primary goal can lead to a more efficient algorithm (Dwork and Roth, 2014, page 211).

2.4.2 Laplace mechanism

Differentially private mechanisms that involve adding random noise utilise various known probability distributions. A common method for achieving $(\epsilon, 0)$-differential privacy is the Laplace mechanism which was originally introduced by Dwork et al. (2006). It can be used to perturb numerical data queries by adding noise drawn from the Laplace distribution. In this work, the Laplace mechanism is applied in output perturbation setting where noise is added to the computed statistics before their release. The representation of the mechanism follows Dwork and Roth (2014, Section 3.3) but multiset notation is used instead of histogram notation.

The Laplace distribution is defined as follows (Dwork and Roth, 2014, Definition 3.2):

Definition 2.13. *(Laplace distribution)* A random real-valued variable Y has a zero-centered Laplace distribution $Y \sim \text{Laplace}(0, b)$ if its probability density function is

$$p(Y) = \frac{1}{2b} \exp\left(-\frac{|Y|}{b}\right),$$

where $b > 0$ is the scale of the distribution.

The used Laplace distribution needs to be adjusted to the query function in order to satisfy the differential privacy definition and to avoid deteriorating the accuracy by adding too much noise. The ℓ_1-sensitivity of a function tells how much the function values can at most change if the input data set is replaced with a neighbouring one (Dwork and Roth, 2014, Definition 3.1).

Definition 2.14. *(ℓ_1-sensitivity)* The ℓ_1-sensitivity of a function $f : \mathcal{D} \to \mathbb{R}^d$ is

$$\Delta f = \max_{\substack{x, y \in \mathcal{D}, \\ \text{dist}(x, y) \leq 2, \\ |x| = |y|}} \|f(x) - f(y)\|_1 .$$

The Laplace mechanism presented as Algorithm 1 (Dwork and Roth, 2014, Definition 3.3) below takes in the data, the query function and the set privacy parameter, computes the results of the query, draws independent and identically distributed (i.i.d.) noise samples from the determined Laplace distribution, and releases the noised query results.

The mechanism is proven to provide strict $(\epsilon, 0)$-differential privacy. The proof here follows Dwork and Roth (2014, Theorem 3.6).

Algorithm 1 Laplace mechanism

Input: Data $x \in \mathcal{D}$, function $f : \mathcal{D} \to \mathbb{R}^d$, privacy parameter ϵ

1: Draw d i.i.d. random variables $Y_i \sim \text{Laplace}(0, \frac{\Delta f}{\epsilon})$, $i = 1, \ldots, d$, and denote $Y = (Y_1, \ldots, Y_d)$

2: Compute $f(x)$

3: Add noise $\widetilde{f(x)} = f(x) + Y$

Output: Perturbed function value $\widetilde{f(x)}$

Theorem 2.15. *Algorithm 1 is $(\epsilon, 0)$-differentially private.*

Proof. We assume two neighbouring data sets $x, \hat{x} \in \mathcal{D}$ differ by most one element i.e. $\text{dist}(x, \hat{x}) \leq 2$ and $|x| = |y|$. Let $f : \mathcal{D} \to \mathbb{R}^d$ be an arbitrary function with ℓ_1-sensitivity Δf, and let $z \in \mathbb{R}^d$ be an arbitrary vector. For Algorithm 1, the ratio between probabilities of getting the same output z with neighbouring inputs x and $\hat{x}$ is

$$
\begin{aligned}
\frac{p(f(x) + Y = z)}{p(f(\hat{x}) + Y = z)} &= \frac{p(Y = z - f(x))}{p(Y = z - f(\hat{x}))} \overset{\text{i.i.d.}}{=} \prod_{i=1}^{d} \frac{\frac{\epsilon}{2\Delta f} \exp\left(-\frac{\epsilon |z_i - f(x)_i|}{\Delta f}\right)}{\frac{\epsilon}{2\Delta f} \exp\left(-\frac{\epsilon |z_i - f(\hat{x})_i|}{\Delta f}\right)} \\
&= \prod_{i=1}^{d} \exp\left[\frac{\epsilon}{\Delta f} \left(|z_i - f(\hat{x})_i| - |z_i - f(x)_i|\right)\right] \\
&\leq \prod_{i=1}^{d} \exp\left[\frac{\epsilon}{\Delta f} \left||z_i - f(\hat{x})_i| - |z_i - f(x)_i|\right|\right] \\
&\overset{\Delta\text{-ineq.}}{\leq} \prod_{i=1}^{d} \exp\left[\frac{\epsilon}{\Delta f} |z_i - f(\hat{x})_i - z_i + f(x)_i|\right] \\
&= \prod_{i=1}^{d} \exp\left[\frac{\epsilon}{\Delta f} |f(x)_i - f(\hat{x})_i|\right] \\
&= \exp\left[\frac{\epsilon}{\Delta f} \sum_{i=1}^{d} |f(x)_i - f(\hat{x})_i|\right] \\
&= \exp\left[\frac{\epsilon}{\Delta f} \left\|f(x) - f(\hat{x})\right\|_1\right] \\
&\leq \exp\left[\frac{\epsilon}{\Delta f} \max_{\substack{x, \hat{x} \in \mathcal{D}, \\ \text{dist}(x, \hat{x}) \leq 2, \\ |x| = |\hat{x}|}} \left\|f(x) - f(\hat{x})\right\|_1\right]
\end{aligned}
$$

$$\overset{\text{Def. 2.14}}{=} \exp\left[\frac{\epsilon}{\Delta f}\Delta f\right] = e^{\epsilon},$$

where triangle inequality was used in the second inequality. This is equivalent to $p(f(x) + Y = z) \leq e^{\epsilon} p(f(\hat{x}) + Y = z)$, so by Definition 2.6, Algorithm 1 is $(\epsilon, 0)$-differentially private. □

Note 2.16. As can be seen from the proof, the sensitivity Δf used in defining the appropriate Laplace distribution in Algorithm 1 can be replaced with any number larger than Δf and the differential privacy condition still holds. This means the exact sensitivity of the query function is not necessarily needed and an approximation that is guaranteed to be greater than the sensitivity can be used instead. (This fact will be needed later in the next chapter.)

Chapter 3

Robust private linear regression

3.1 Problem setting

The robust private linear regression algorithm was originally introduced by Honkela et al. (2016). It is designed for applications where a linear relationship is assumed to exist between explanatory and dependent data and the details of the relationship are desired to be learnt in a way that does not risk the privacy of individual data points. The studied model, defined next in Section 3.2, is Bayesian linear regression which assumes the error terms of the linear regression are normally distributed. The used privacy definition is bounded $(\epsilon, 0)$-differential privacy (Definition 2.6 with $\delta = 0$).

The privacy setting and the outline of the algorithm are illustrated in Figure 3.1, and the details will be explained in the next sections, following Honkela et al. (2017). The idea is to deploy offline output perturbation where the data curator keeps the data itself private and only releases specific noise-perturbed statistics computed from the data. The data analyst can then learn the regression coefficients (β) of the linear regression based on the privatised statistics. The learnt coefficients can be used to study the details of how each explanatory variable is linked to the dependent variable and also to predict the future responses for observed explanatory data. A useful property of the robust private linear regression mechanism is that the released statistics of a private data set can easily be combined together with similar statistics computed from other data sets that are assumed to follow the same model but can be of different sizes. This makes it possible to use an additional non-private data set comprising data points that do not require privacy protection. Similarly, the property can be used to combine multiple private data sets that are behind separate privacy walls and curators. This way the mechanism enables e.g. different hospitals to share useful information for data analysis without leaking too much knowledge about any individual patient in their own hospital. Moreover, in the cases where the computation processes themselves require strict privacy, additional

cryptographic methods can be applied (see e.g. Heikkilä et al., 2017).

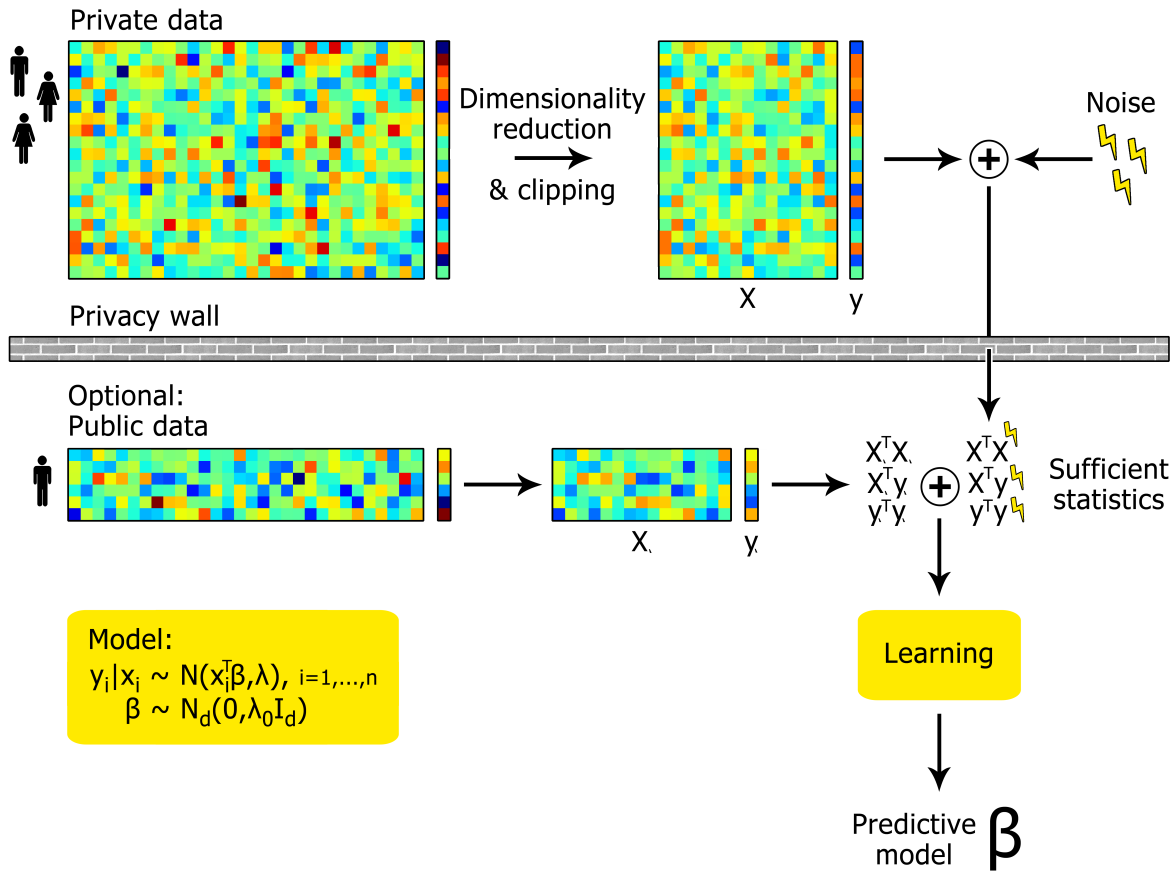

Figure 3.1: **The privacy setting and outline of the robust private linear regression algorithm.** The data curator only releases noise-perturbed statistics computed from the private data. They can be combined with non-perturbed statistics computed from an additional public data set, and the model (3.1) is learnt using the combined statistics.

In differential privacy settings, there is always a trade-off to be paid between the privacy level and the accuracy of data analysis results, and cleverly designed privacy mechanisms should seek to minimise the paid cost. In order to do this, the presented algorithm deploys two crucial steps to reduce the amount of required noise: dimensionality reduction of the data, and constricting the data points inside some known bounds by clipping the values with specified thresholds. The details of these methods are discussed in the following sections.

3.2 Bayesian linear regression

The Bayesian linear regression model (without an intercept term) is

$$\begin{aligned} y_i|x_i &\sim N(x_i^T\beta, \lambda) \text{ for } i = 1, \ldots, n \\ \beta &\sim N_d(0, \lambda_0 I_d), \end{aligned} \tag{3.1}$$

where $x_i \in \mathbb{R}^d$ are the n observed samples of d predictor variables, $y_i \in \mathbb{R}$ are the n observed samples of the target variable and assumed to be i.i.d., and the elements of $\beta \in \mathbb{R}^d$ are the regression coefficients. The parameters $\lambda_0 > 0$ and $\lambda > 0$ are the precision parameters of the normal and multivariate normal distributions, and act as regularisers: the parameter λ_0 controls the magnitude of the regression coefficients, and the paramater λ controls how strictly the target variables obey a linear relationship to the predictors. I_d denotes the $d \times d$ identity matrix. In other words, the model assumes the dependent variable y_i has a linear relationship to the explanatory variables in vector x_i and the unobserved errors follow a normal distribution with zero mean and variance $1/\lambda$. The regression coefficients are assumed to follow a zero-centered multivariate normal distribution with a diagonal covariance matrix $(\lambda_0 I_d)^{-1} = (1/\lambda_0) I_d$.

Denote the design matrix $X = [x_1^T, \ldots, x_n^T]^T \in \mathbb{R}^{n \times d}$, where rows correspond to samples and columns correspond to predictor variables or features, and denote $y = [y_1, \ldots, y_n]^T \in \mathbb{R}^n$.

The probability density function of the prior distribution of β is

$$\begin{aligned} p(\beta) &= (2\pi)^{-\frac{d}{2}} |\lambda_0^{-1} I_d|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\beta^T(\lambda_0 I_d)\beta\right) \\ &= (2\pi)^{-\frac{d}{2}} (\lambda_0^{-d})^{-\frac{1}{2}} \exp\left(-\frac{\lambda_0}{2}\beta^T\beta\right) \\ &= \left(\frac{\lambda_0}{2\pi}\right)^{\frac{d}{2}} \exp\left(-\frac{\lambda_0}{2}\beta^T\beta\right). \end{aligned} \tag{3.2}$$

Because the samples y_i are assumed to be i.i.d., they follow the multivariate normal distribution $N_n(X\beta, \lambda I_n)$. The likelihood function of the data $D = (X, y)$ is then

$$\begin{aligned} p(y|X, \beta) &= (2\pi)^{-\frac{n}{2}} |\lambda^{-1} I_n|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(y - X\beta)^T \lambda I_n (y - X\beta)\right) \\ &= (2\pi)^{-\frac{n}{2}} (\lambda^{-n})^{-\frac{1}{2}} \exp\left[-\frac{\lambda}{2}\left(y^T - \beta^T X^T\right)(y - X\beta)\right] \\ &= \left(\frac{\lambda}{2\pi}\right)^{\frac{n}{2}} \exp\left[-\frac{\lambda}{2}\left(y^T y - y^T X\beta - \beta^T X^T y + \beta^T X^T X\beta\right)\right] \end{aligned}$$

$$
\begin{aligned}
&= \left(\frac{\lambda}{2\pi}\right)^{\frac{n}{2}} \exp\left[-\frac{\lambda}{2}\left(y^T y - (\beta^T X^T y)^T - \beta^T X^T y + \beta^T X^T X \beta\right)\right] \\
&= \left(\frac{\lambda}{2\pi}\right)^{\frac{n}{2}} \exp\left[-\frac{\lambda}{2}\left(y^T y - 2\beta^T X^T y + \beta^T X^T X \beta\right)\right], \qquad (3.3)
\end{aligned}
$$

where the last equality follows from the fact that $\beta^T X^T y$ is a scalar and therefore $(\beta^T X^T y)^T = \beta^T X^T y$.

Assuming the parameters λ and λ_0 are known, and using (3.2), (3.3) and the Bayes' theorem, the probability density function of the posterior distribution of β can be written as

$$
\begin{aligned}
p(\beta|D) &= \frac{p(\beta)p(D|\beta)}{p(D)} \propto p(\beta)p(D|\beta) \\
&= \left(\frac{\lambda_0}{2\pi}\right)^{\frac{d}{2}} \left(\frac{\lambda}{2\pi}\right)^{\frac{n}{2}} \exp\left[-\frac{1}{2}\left(\lambda_0 \beta^T \beta + \lambda \beta^T X^T X \beta - \lambda 2 \beta^T X^T y + \lambda y^T y\right)\right] \\
&\propto \exp\left[-\frac{1}{2}\left(\beta^T(\lambda_0 I_d + \lambda X^T X)\beta - 2\beta^T \lambda X^T y\right)\right]. \qquad (3.4)
\end{aligned}
$$

Denote $\Lambda_* = \lambda_0 I_d + \lambda X^T X$. Note that

$$
\begin{aligned}
\left[(\lambda X^T y)^T (\Lambda_*^{-1})^T \Lambda_* \beta\right]^T &= [(\Lambda_*^{-1})^T \Lambda_* \beta]^T [(\lambda X^T y)^T]^T \\
&= [\Lambda_* \beta]^T [(\Lambda_*^{-1})^T]^T \lambda X^T y \\
&= \beta^T \Lambda_*^T \Lambda_*^{-1} \lambda X^T y \\
&= \beta^T \Lambda_* \Lambda_*^{-1} \lambda X^T y \\
&= \beta^T \lambda X^T y,
\end{aligned}
$$

so because $\beta^T \lambda X^T y$ is a scalar, it holds that $\beta^T \lambda X^T y = (\lambda X^T y)^T (\Lambda_*^{-1})^T \Lambda_* \beta$. Therefore, (3.4) can be written as

$$
\begin{aligned}
p(\beta|D) &\propto \exp\left[-\frac{1}{2}\left(\beta^T \Lambda_* \beta - \beta^T \lambda X^T y - (\lambda X^T y)^T (\Lambda_*^{-1})^T \Lambda_* \beta\right)\right] \\
&\propto \exp\left[-\frac{1}{2}\left(\beta^T \Lambda_* \beta - \beta^T \lambda X^T y - (\lambda X^T y)^T (\Lambda_*^{-1})^T \Lambda_* \beta + (\lambda X^T y)^T (\Lambda_*^{-1})^T \lambda X^T y\right)\right] \\
&= \exp\left[-\frac{1}{2}\left(\beta^T - (\lambda X^T y)^T (\Lambda_*^{-1})^T\right)\left(\Lambda_* \beta - \lambda X^T y\right)\right] \\
&= \exp\left[-\frac{1}{2}\left(\beta - \Lambda_*^{-1}(\lambda X^T y)\right)^T \Lambda_* \left(\beta - \Lambda_*^{-1}(\lambda X^T y)\right)\right]
\end{aligned}
$$

$$= \exp\left[-\frac{1}{2}(\beta-\mu_*)^T\Lambda_*(\beta-\mu_*)\right],$$

when denoting $\mu_* = \Lambda_*^{-1}(\lambda X^T y)$. Since the probability density function must integrate to unity, the posterior is the normal distribution $N_d(\mu_*, \Lambda_*)$ with the probability density function

$$p(\beta|D) = (2\pi)^{-\frac{d}{2}}|\Lambda_*^{-1}|^{-\frac{1}{2}}\exp\left[-\frac{1}{2}(\beta-\mu_*)^T\Lambda_*(\beta-\mu_*)\right].$$

The minimum mean square error (MMSE) estimator for the regression coefficients β is therefore the posterior mean

$$\mu_* = (\lambda_0 I_d + \lambda X^T X)^{-1}(\lambda X^T y).$$

Since the posterior is Gaussian, the posterior mode is the same as the posterior mean so μ_* is also the maximum a posteriori (MAP) estimate solution. The learnt solution μ_* can be used to predict the output y_i for x_i as

$$\hat{y}_i = x_i^T \mu_*. \tag{3.5}$$

Hence, if the parameters λ, λ_0 are assumed to be known as in Honkela et al. (2016), the two statistics $X^T X = \sum_{i=1}^n x_i x_i^T \in \mathbb{R}^{d\times d}$ and $X^T y = \sum_{i=1}^n x_i y_i \in \mathbb{R}^d$ computed from the data determine the posterior distribution and the MMSE and MAP solution. Here they are therefore called sufficient statistics, as they are enough to determine the posterior and hence enough to know for a data analyst who wants to make inference about the regression coefficients β based on this model. They can therefore be used in a differentially private output perturbation mechanism: the data curator can keep private the actual data X, y and only release noise-perturbed versions of the sufficient statistics $X^T X$ and $X^T y$. This is the basic idea the original version of the robust private linear regression mechanism was based on (Honkela et al., 2016).

3.2.1 Priors for the precision parameters

Instead of fixing the values of the precision parameters λ, λ_0, a more robust alternative is to assign them prior distributions, which is the idea in the new version of the proposed mechanism (Honkela et al., 2017). A natural choice is to use the conjugate prior, the gamma distribution, whose support is the set of positive real numbers.

$$\begin{aligned} \lambda_0 &\sim \text{Gamma}(a_0, b_0) \\ \lambda &\sim \text{Gamma}(a, b) \end{aligned} \tag{3.6}$$

The probability density function of the Gamma(a, b) distribution with shape parameter $a > 0$ and rate parameter $b > 0$ at point $\lambda > 0$ is

$$p(\lambda) = \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda},$$

where Γ denotes the gamma function.

Computing the posterior mean analytically is infeasible in this setting. Instead, one can use computational Markov chain Monte Carlo (MCMC) methods (Gamerman and Lopes, 2006) to draw samples $\beta^{(k)}$ from the posterior distribution of β. Another alternative is to use automatic differentation variational inference (ADVI) (Kucukelbir et al., 2017), where a variational distribution is fitted to the posterior and samples $\beta^{(k)}$ are drawn from the fitted distribution. Since the samples are distributed (approximately) as the posterior distribution, the average over a large number of samples gives the posterior mean which is the desired solution to the linear regression task. The predicted output for input x_i using m samples $\beta^{(k)}$ from the posterior or fitted distribution is

$$\hat{y}_i = \sum_{k=1}^{m} x_i^T \beta^{(k)}. \tag{3.7}$$

In order to do sampling using MCMC or ADVI, the data log-likelihood is needed. Taking the logarithm of the likelihood (3.3) gives

$$\log p(D|\beta) = \frac{n}{2} \log \left(\frac{\lambda}{2\pi} \right) - \frac{\lambda}{2} \left(\beta^T X^T X \beta - 2\beta^T X^T y + y^T y \right). \tag{3.8}$$

As can be seen above, now the two statistics $X^T X$ and $X^T y$ alone are not sufficient: a third statistic $y^T y = \sum_{i=1}^{n} y_i^2 \in \mathbb{R}$ is needed as well. The new version of the robust private linear regression mechanism (Honkela et al., 2017) is based on these three sufficient statistics $X^T X$, $X^T y$, and $y^T y$, and sampling using ADVI. The sample size n is also needed in the likelihood computation, and since the mechanism is based on bounded differential privacy (Definition 2.6), it is clear the exact sample size can be released without compromising the privacy.

3.3 Differentially private mechanism

3.3.1 Perturbation of sufficient statistics

In order to guarantee differential privacy as defined in Definition 2.6, each of the three sufficient statistics is independently perturbed with carefully tailored noise. For this

purpose, the robust private linear regression algorithm applies the Laplace mechanism introduced in Section 2.4.2.

The perturbation of each sufficient statistic needs to be done independently and in a differentially private manner. The composition theorem (Theorem 2.12) then guarantees that releasing all three perturbed statistics together also satisfies the definition of differential privacy: We assume the privacy budget $\epsilon > 0$ is divided into three portions $p_1\epsilon, p_2\epsilon, p_3\epsilon$, where $p_1, p_2, p_3 > 0$ and $p_1 + p_2 + p_3 = 1$. Now, if the release of the perturbed version of the input covariance term $X^T X$ is guaranteed to be $p_1\epsilon$-differentially private, the release of the perturbed version of the target term $X^T y$ is $p_2\epsilon$-differentially private, and the release of the perturbed version of the output covariance term $y^T y$ is p_3-differentially private, the composition theorem then states that the release of the three statistics is differentially private with the privacy parameter value $p_1\epsilon + p_2\epsilon + p_3\epsilon = \epsilon$. The non-trivial task of choosing the optimal privacy budget split is discussed in Section 3.4.1.

Since each of the sufficient statistics $X^T X \in \mathbb{R}^{d\times d}$, $X^T y \in \mathbb{R}^d$, $y^T y \in \mathbb{R}$ and their perturbed versions are of fixed sizes independent from the sample size n of the data set, the privacy mechanism does not need to be adjusted according to the data set size. The same mechanism can therefore be used to privatise the sufficient statistics computed from multiple data sets of varying sample sizes as long as the dimensionality d is the same.

Combining the sufficient statistics computed from two different data sets is also easy as the corresponding statistics can simply be added together, which for non-perturbed versions can easily be shown to produce the same result as computing the sufficient statistics of the pooled data set: Let $A \in \mathbb{R}^{n_A\times d}, a \in \mathbb{R}^{n_A}, B \in \mathbb{R}^{n_B\times d}, b \in \mathbb{R}^{n_B}$ and be arbitrary and let

$$C = \left[\frac{A}{B}\right] \in \mathbb{R}^{(n_A+n_B)\times d}, c = \left[\frac{a}{b}\right] \in \mathbb{R}^{n_A+n_B}$$

denote the vertical concatenations of the two matrices and vectors. Now $C^T = \left[A^T||B^T\right]$ and for any $(i, j) \in \{1, \ldots, d\}^2$ it holds that

$$[C^T C]_{ij} = \sum_{k=1}^{n_A} A_{ki}A_{kj} + \sum_{k=1}^{n_B} B_{ki}B_{kj} = [A^T A + B^T B]_{ij},$$

and therefore $C^T C = A^T A + B^T B$. In other words, adding together the corresponding (input or output) covariance matrices of two different data sets results in the same matrix as computing the covariance matrix of the combined data set. Similarly, $c^T = [a^T||b^T]$ and for any $i \in \{1, \ldots, d\}$ it holds that

$$[C^T c]_i = \sum_{i=1}^{n_A} A_{ki}a_k + \sum_{i=1}^{n_B} B_{ki}b_k = [A^T a + B^T b]_i,$$

so $C^T c = A^T a + B^T b$. Thus, the target terms of two data sets can also be added together and the resulting vector is the same as the target term computed from the combined data set. By induction, the corresponding sufficient statistics of any number of data sets can simply be added together. For private data sets, noise is added to the statistics before their release, so in this combining procedure, the noise are also added together.

3.3.2 Data projection

A crucial step in many differentially private mechanisms is constricting the data inside some known bounds: Since the Definition 2.6 of differential privacy requires the probability of any output should stay almost the same with any possible neighbouring data sets, the amount of added noise in perturbation should be able to mask all possible values in the data. Therefore, suitably bounding the data reduces the amount of required noise. The standard way to do this is to linearly transform the data inside some defined bounds (Zhang et al., 2012). However, the disadvantage of this method is that it may produce a very small scale for most of the data in case a few far away outliers happen to exist in the data set. In the robust private linear regression mechanism, an alternative method called projection or clipping is introduced (Honkela et al., 2016): the bounds are first chosen so that they cover the essential variation in the data, and any outlying data points outside these bounds are then projected inside them. In this way, the scale of any meaningful variation does not get diluted and the bounds are chosen independent from the scale of the outliers. The possible outliers get eliminated while the rest of the data experience little changes, which makes the mechanism more robust — hence the name of the algorithm.

3.3.3 Formal definition of the algorithm

The mechanism of releasing the three privatised sufficient statistics consists of projecting the data, computing the sufficient statistics, and adding suitable noise to the statistics. The mechanism is presented in detail in Algorithm 2 and proven to be differentially private in Theorem 3.9.

Theorem 3.9. *Algorithm 2 is* $(\epsilon, 0)$*-differentially private.*

Proof. We assume two neighbouring data sets D_1, D_2 of the same size are clipped using the same clipping thresholds B_x, B_y, thus producing clipped design matrices $Z, \hat{Z} \in \mathbb{R}^{n \times d}$ that differ at most one row, and clipped target vectors $u, \hat{u} \in \mathbb{R}^n$ that can only differ at the one element corresponding to the differing row in $Z, \hat{Z}$. The two versions of the possibly differing row are denoted by $v, \hat{v} \in \mathbb{R}^d$ and the two versions of the possibly differing element in the target vectors are denoted by $w, \hat{w} \in \mathbb{R}$. The noise perturbation mechanism comprises three independent parts, each of which is now proven to be differentially private.

Algorithm 2 Differentially private sufficient statistics

Input: Data design matrix $X \in \mathbb{R}^{n \times d}$, target data vector $y \in \mathbb{R}^n$, clipping thresholds B_x, B_y, privacy parameter ϵ, privacy budget splitting proportions $p_1, p_2, p_3 > 0$ that satisfy $p_1 + p_2 + p_3 = 1$

for $i = 1$ to n **do**
  for $j = 1$ to d **do**
    $Z_{ij} = \max(-B_x, \min(B_x, X_{ij}))$
  end for
  $u_i = \max(-B_y, \min(B_y, y_i))$
end for
for $i = 1$ to n **do**
  for $j = i$ to d **do**
    $P_{ij} \sim \text{Laplace}\left(0, \frac{d(d+1)B_x^2}{p_1 \epsilon}\right)$
    $P_{ji} = P_{ij}$
  end for
end for
for $i = 1$ to n **do**
  $Q_i \sim \text{Laplace}\left(0, \frac{2dB_xB_y}{p_2 \epsilon}\right)$
end for
$R \sim \text{Laplace}\left(0, \frac{B_y^2}{p_3 \epsilon}\right)$
$S_{xx} = Z^T Z + P$
$S_{xy} = Z^T u + Q$
$S_{yy} = u^T u + R$

Output: Perturbed sufficient statistics S_{xx}, S_{xy}, S_{yy}

(i) $S_{xx} = Z^T Z + P$ is $p_1\epsilon$-differentially private.

Since the input covariance matrix $Z^T Z \in \mathbb{R}^{d \times d}$ is symmetric, it contains $\frac{d(d+1)}{2}$ unique elements in its upper diagonal, and the elements in the lower diagonal are equal to their opposite elements on the other side of the diagonal. Therefore, the number of noise samples required to perturb all unique elements is $\frac{d(d+1)}{2}$.

By Definition 2.14, the ℓ_1-sensitivity of the matrix $Z^T Z$ is

$$\begin{aligned}
\Delta(Z^T Z) &= \max_{\substack{\text{dist}(D_1, D_2) \leq 2, \\ |D_1| = |D_2|}} \left\| Z^T Z - \hat{Z}^T \hat{Z} \right\|_1 \\
&= \max_{\substack{\text{dist}(D_1, D_2) \leq 2, \\ |D_1| = |D_2|}} \sum_{i=1}^{d} \sum_{j=i}^{d} \left| (Z^T Z)_{ij} - (\hat{Z}^T \hat{Z})_{ij} \right|
\end{aligned}$$

$$= \max_{\substack{\mathrm{dist}(D_1,D_2)\le 2,\\ |D_1|=|D_2|}} \sum_{i=1}^{d}\sum_{j=i}^{d}\left|\sum_{k=1}^{n} Z_{ki}Z_{kj} - \sum_{k=1}^{n}\hat{Z}_{ki}\hat{Z}_{kj}\right|$$

$$= \max_{\substack{\mathrm{dist}(D_1,D_2)\le 2,\\ |D_1|=|D_2|}} \sum_{i=1}^{d}\sum_{j=i}^{d}|v_iv_j - \hat{v}_i\hat{v}_j| \tag{3.10}$$

By fixing the data sets D_1, D_2 and the corresponding elements $v, \hat{v}$ so that the expression (3.10) gets its maximal value, it can be evaluated as

$$\begin{aligned}\sum_{i=1}^{d}\sum_{j=i}^{d}|v_iv_j - \hat{v}_i\hat{v}_j| &\le \sum_{i=1}^{d}\sum_{j=i}^{d}(|v_i||v_j| + |\hat{v}_i||\hat{v}_j|) \le \sum_{i=1}^{d}\sum_{j=i}^{d}(B_x^2 + B_x^2)\\ &= \frac{d(d+1)}{2}\cdot 2\cdot B_x^2 = d(d+1)B_x^2.\end{aligned}$$

Therefore, for the ℓ_1-sensitivity of Z^TZ it holds that $\Delta(Z^TZ) \le d(d+1)B_x^2$. By the proof of Theorem 2.15, drawing $\frac{d(d+1)}{2}$ i.i.d. samples from Laplace $\left(0, \frac{d(d+1)B_x^2}{p_1\epsilon}\right)$, constructing a symmetric noise matrix P from these samples as in Algorithm 2, and releasing the perturbed input covariance matrix $S_{xx} = Z^TZ + P$ is $p_1\epsilon$-differentially private.

(ii) $S_{xy} = Z^Tu + Q$ is $p_2\epsilon$-differentially private.

The sensitivity of the clipped target term computation is

$$\begin{aligned}\Delta(Z^Tu) &= \max_{\substack{\mathrm{dist}(D_1,D_2)\le 2,\\ |D_1|=|D_2|}} \left\|Z^Tu - \hat{Z}^T\hat{u}\right\|_1\\ &= \max_{\substack{\mathrm{dist}(D_1,D_2)\le 2,\\ |D_1|=|D_2|}} \sum_{j=1}^{d}\left|(Z^Tu)_j - (\hat{Z}^T\hat{u})_j\right|\\ &= \max_{\substack{\mathrm{dist}(D_1,D_2)\le 2,\\ |D_1|=|D_2|}} \sum_{j=1}^{d}\left|\sum_{i=1}^{n} Z_{ij}u_i - \sum_{i=1}^{n}\hat{Z}_{ij}\hat{u}\right|\\ &= \max_{\substack{\mathrm{dist}(D_1,D_2)\le 2,\\ |D_1|=|D_2|}} \sum_{j=1}^{d}|v_jw - \hat{v}_j\hat{w}|\\ &= d\cdot \max_{\substack{\mathrm{dist}(D_1,D_2)\le 2,\\ |D_1|=|D_2|}} |v_jw - \hat{v}_j\hat{w}|\\ &= 2dB_xB_y,\end{aligned}$$

where the last equality follows from the fact that $|v_j w - \hat{v}_j \hat{w}| \leq |v_j||w| + |\hat{v}_j||\hat{w}| \leq B_x B_y + B_x B_y = 2B_x B_y$ for all possible values of $v_j, \hat{v}_j, w, \hat{w}$, and by choosing $v_j = B_x$, $\hat{v}_j = -B_x$ for every $j = 1, \ldots, d$ and $w = B_y$, $\hat{w} = B_y$, the upper bound can be reached as $|v_j w - \hat{v}_j \hat{w}| = |B_x B_y - (-B_x) B_y| = 2B_x B_y$.

Therefore, since the noise samples Q_i, $i = 1, \ldots, d$, in Algorithm 2 are i.i.d. drawn from the Laplace distribution Laplace$(0, \frac{2dB_x B_y}{p_2 \epsilon})$, Theorem 2.15 says that releasing the perturbed target term $S_{yy} = Z^T u + Q$ is $p_2\epsilon$-differentially private.

(iii) $S_{yy} = u^T u + R$ is $p_3\epsilon$-differentially private.

The ℓ_1-sensitivity of the output covariance term $u^T u$ is

$$\begin{aligned}
\Delta(u^T u) &= \max_{\substack{\text{dist}(D_1, D_2) \leq 2, \\ |D_1| = |D_2|}} \left\| u^T u - \hat{u}^T \hat{u} \right\|_1 \\
&= \max_{\substack{\text{dist}(D_1, D_2) \leq 2, \\ |D_1| = |D_2|}} \left| \sum_{i=1}^{n} u_i^2 - \sum_{i=1}^{n} \hat{u}_i^2 \right| \\
&= \max_{\substack{\text{dist}(D_1, D_2) \leq 2, \\ |D_1| = |D_2|}} \left| w^2 - \hat{w}^2 \right| \\
&= B_y^2,
\end{aligned}$$

where the last equality follows from the fact that $|w^2 - \hat{w}^2| \leq \max\{w^2, \hat{w}^2\} \leq B_y^2$ for all possible values of $w, \hat{w}$, and the upper bound can be reached by setting $w = B_y$, $\hat{w} = 0$.

Thus, by Theorem 2.15, drawing a noise sample R from Laplace$(0, \frac{B_y^2}{p_3 \epsilon})$ and releasing the perturbed output covariance term $S_{yy} = u^T u + R$ is $p_3\epsilon$-differentially private.

Because $p_1\epsilon + p_2\epsilon + p_3\epsilon = \epsilon$, by Corollary 2.12 and (i)-(iii), releasing the sufficient statistics S_{xx}, S_{xy} and S_{yy} together by Algorithm 2 is ϵ-differentially private. □

3.4 Determining the privacy budget split and the projection thresholds

The choices for the values of the privacy budget proportions p_1, p_2, p_3 and the projection thresholds B_x, B_y are crucial to the prediction performance of the algorithm. As seen in earlier work and noted by Honkela et al. (2016), the clipping thresholds strongly depend on the data set size. An effective method is to generate a synthetic auxiliary data set of the same size as the data set to be studied and select the parameter values that lead

to the best prediction performance on the auxiliary data. The auxiliary data set of n samples and dimensionality d is generated according to a model similar to (3.1):

$$\begin{aligned} x_i &\sim N(0, I_d) \text{ for } i = 1, \ldots, n \\ y_i|x_i &\sim N(x_i^T \beta, \lambda) \text{ for } i = 1, \ldots, n \\ \beta &\sim N(0, \lambda_0 I_d). \end{aligned} \tag{3.11}$$

3.4.1 Privacy budget split

The simplest choice would be to spend an equal proportion of the privacy budget on each sufficient statistic. It was used in the older version of the robust private linear regression algorithm that spent half of the privacy budget on the term $X^T X$ and the remaining half on the term $X^T y$ (Honkela et al., 2016). However, it is likely not the optimal solution. Considering the term $X^T X$ is composed of $\frac{d(d+1)}{2}$ unique elements and $X^T y$ of d elements while $y^T y$ is just a scalar, it is sensible to doubt they would be equally important to the analysis and that their utility would suffer from the added noise in the same way. Moreover, $X^T y$ is the only term that in itself contains information about both the explanatory and dependent variables and thus potentially holds a more special position in the linear regression task than the others. Therefore, it is possibly better to spend more of the privacy budget on an especially important term and thus reduce the amount of noise added to it. In the new version of the algorithm (Honkela et al., 2017), the privacy budget split is optimised using an auxiliary data method that is similar in nature to the method used to determine the projection thresholds (Honkela et al., 2016).

The optimal budget split has to be decided before choosing the clipping thresholds because the method of determining the clipping thresholds requires generating the correct amount of noise for each sufficient statistic. First, an auxiliary data set of the same size as the actual data set is generated according to the model (3.11). All possible combinations of budget split proportions $\{p_1, p_2, p_3\} \in \{0.05, 0.1, 0.15, \ldots, 0.90\}^3$, where $p_1 + p_2 + p_3 = 1$, are studied, and for each split, the optimal clipping thresholds are decided by a method explained in the Section 3.4.2. As in Algorithm 2, the auxiliary data are then projected using the acquired thresholds, and the sufficient statistics are computed from the clipped data and perturbed according to the current privacy budget split. The model is learnt and sampled using ADVI and the prediction is computed as in (3.7). The prediction accuracy is then evaluated between the prediction and the actual values as in (3.14). In practice, the prediction performance should be averaged over several auxiliary data sets and perturbation noise samples. The privacy budget split leading to the best accuracy is chosen and used in all tests.

3.4.2 Projection thresholds

Given a privacy budget split, the optimal clipping thresholds are chosen in a similar way by first generating an auxiliary data set of the same size as the actual data (Honkela et al., 2016). The clipping thresholds are parameterised as functions of the data standard deviations as

$$B_x = \omega_x \sigma_x, B_y = \omega_y \sigma_y \tag{3.12}$$

$$\{\omega_x, \omega_y\} \in \{0.1\omega\}_{\omega=1}^{20}, \tag{3.13}$$

where σ_x and σ_y denote the standard deviations of X (considering all dimensions) and y of the auxiliary data set. The prediction accuracy with each of these pairs (B_x, B_y) is then studied. As in Algorithm 2, the auxiliary data are projected using the current clipping thresholds, and the sufficient statistics are computed and perturbed. The model is fitted and sampled with ADVI and the prediction is evaluated as in (3.7). Alternatively, to save time, the analytical posterior can be learnt as in (3.2) and the prediction computed as (3.5) using fixed values for the precision parameters λ, λ_0. The performance is evaluated between the prediction and the actual values as in (3.14). The results should be averaged over several auxiliary data sets and noise samples. The pair (ω_x, ω_y) leading to the best prediction performance is then chosen to be used with the actual data set. The final projection thresholds are defined as in (3.12) using the corresponding standard deviations of the actual data set.

3.5 Pre-processing

The dimensionality of the data is a substantial problem in differentially private noise-perturbation in the sense that it can potentially force the required noise level so high it completely destroys the accuracy in data analysis. Hence, in order to reduce the amount or required noise, a necessary step is to apply some kind of a dimensionality reduction method on high-dimensional data. In the presented mechanism, the selected method is to pick the dimensions that are assumed or known to be the most predictive of the dependent variable and discard all other, less relevant dimensions.

The reduced data are then normalised by first removing the mean from each remaining dimension (column) in the design matrix X. Each data point (row) is then normalised to unit length in terms of L_2-norm in order to equalise the effect of each point. The mean is also removed from the target vector y. If necessary, rows containing missing values are simply dropped out of the data set.

3.6 Running order

The steps of the algorithm described above are executed in the following order:

I Parameter choices

i) Privacy budget split p_1, p_2, p_3

ii) Projection threshold parameters ω_x, ω_y

II Data pre-processing

i) Dimensionality reduction down to d

ii) Normalisation

iii) Elimination of missing values

III Release of differentially private statistics

i) Data projection

ii) Sufficient statistics $X^T X, X^T y, y^T y$

iii) Noise perturbation of each sufficient statistic

IV Analysis

i) Bayesian linear regression model fitting

ii) Prediction

Step I requires knowing the size and (reduced) dimensionality of the data set but otherwise steps I and II are independent of each other. Thus, either of them can be performed as the first step.

3.7 Data

3.7.1 Synthetic data

In order to demonstrate the performance of the algorithm on some simple data as a sanity check, a synthetic data set following the model (3.11) is generated with $n = 1000$ samples, dimensionality of $d = 10$, and precision parameter values $\lambda = \lambda_0 = 1$. As the dimensionality of the data is kept relatively low, it is not reduced further down. The data also do not contain missing values. The data are otherwise pre-processed as explained in Section 3.5.

3.7.2 Drug sensitivity data

As a use case, a drug sensitivity prediction task is studied in the paper Honkela et al. (2017). The data are from the *Genomics of Drug Sensitivity in Cancer* (GDSC) project introduced by Yang et al. (2013) and Garnett et al. (2012) (data release 6.1, March 2017, downloaded from `http://www.cancerrxgene.org`). The ongoing project provides the currently largest and still growing public data set for studying the drug response in cancer cells (Yang et al., 2013). The aim of the research is to discover new biomarkers of drug response: measurable genomic characteristics that indicate the cancer cells' sensitivity to various drugs (Garnett et al., 2012). As noted by Garnett et al. (2012), single gene mutations can indicate sensitivity to certain drugs but are usually not enough to fully explain the observed drug response. Instead, the response can depend on complex relations between a large number of genes.

The subset of the studied GDSC data consists of the gene expression data of 985 human cancer cell lines and their corresponding responses to 265 drugs. The gene expression data measured using DNA microarrays indicates which genes are actively being used in protein and RNA synthesis (being expressed) (Baldi and Hatfield, 2002, Preface). The goal is to fit the presented linear regression model (3.1) to the data and predict which cancer cell lines are sensitive to which drugs and which are not.

Naveed et al. (2015) overview the recent rapid progress in the field of genomic information analysis: The genome sequencing technology has quickly become increasingly more accurate and affordable, which has resulted in an ever-growing amount of collected genomic data. Improved technology and sufficiently large quantities of data together make it possible to conduct more profound and precise analysis and to develop new, better medical treatments. They also enable more extensive use of personalised medicine: tailoring treatments based on a patient's individual genetic make-up. The biomedical scientists see great potential and benefits in these advancements. However, collecting, storing, and releasing genomic information also pose privacy risks as discussed by Naveed et al. (2015) who in their article aim to compile essential knowledge about the special characteristics, risks, and strategies related to genomic privacy: Genomic data can potentially reveal an illness or a predisposition to one, a link between relatives, certain characteristics related to the individual's appearance or behaviour, and a recent interaction between people whose DNA are found in the vicinity of each other. As genes can identify a person and mostly stay constant throughout their life, the privacy of genomic data will always stay important. In fact, genomic privacy will likely become increasingly more vital due to the continuous development of genetics which will make it possible to infer more and more about an individual based on their genetic make-up.

The motivation for the application of differentially private mechanisms to the drug sensitivity prediction task is thus evident: Since gene expression data can identify the in-

dividual and even summary statistics pose the risk of exposing information, releasing this kind of data can potentially expose e.g. the medical statuses of involved patients. On the other hand, researchers desire to acquire vast amounts of empirical data that allow them to properly study the effectiveness of drugs and to develop better ones. Therefore, differentially private methods that sufficiently mask the presence of each individual patient's data while also allowing good data utility are highly valuable.

The dimensionality of the RMA-normalised gene expression data in the GDSC data set is high, $d = 17490$. It is reduced down using expert knowledge about genes that are frequently mutated in cancer cells, as provided by the GDSC project. The information about mutations was originally retrieved from `http://www.cancerrxgene.org/translation/Gene` as stated by Honkela et al. (2016) but this URL is no longer accessible. The mostly corresponding information can be found at `http://www.cancerrxgene.org/translation/Feature`. The genes are ordered based on the mutation counts acquired from the COSMIC database at `http://cancer.sanger.ac.uk/cosmic/curation` and the 64 genes with most mutations are picked for further analysis. The drug sensitivity results are measured by log-transformed half maximal inhibitory concentration values (IC50) (Garnett et al., 2012), indicating the drug concentration that causes 50% inhibition in cancer cells. The IC50 values in the data set were determined from curves fitted to dose response data that was measured at nine different drug concentrations (Garnett et al., 2012). The gene expression data and drug response data are further processed as explained in Section 3.5.

3.8 Evaluation

The prediction performance of the presented robust private linear regression algorithm is evaluated on both the synthetic data and the GDSC data set. Since the two data sets are approximately the same size, the same privacy budget split, as represented in Section 4.1, is used for both sets.

3.8.1 Synthetic data

As a sanity check, the algorithm is tested on a generated synthetic data set and compared with two modified, non-private versions of the same algorithm that do not add noise to the sufficient statistics: one version with clipping and one with no clipping. The mechanism is tested in settings where the number of available private data is varied in $n_{pv} = 0, 100, 200, \ldots, 800$, and the privacy parameter is set to $\epsilon = 2$. The prediction performance of the private algorithm is also evaluated on $n_{pv} = 500$ private samples as a function of the privacy budget that is varied in $\epsilon = 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0$. In each

test, a small additional non-private data set of $n_{npv} = 10$ samples is used.

In each test, the data are split into training and test sets in a 50-fold Monte Carlo cross-validation procedure. In each split, the first $n_{test} = 100$ samples are used as a validation set and the rest are split into private and non-private training sets of defined sizes. The model is trained on the training data, and the prediction on the validation data is computed using the fitted model as in (3.7). The accuracy of the prediction is calculated, and the final results are computed as the averages over all cross-validation folds.

The accuracy of the model prediction is evaluated using Spearman's rank correlation coefficient which is explained by Siegel (1956, page 202): The coefficient measures the rank correlation between two variables that are measured in such a scale that the values can be ordered according to some criterion. Given a variable X and an ordered set of data samples, the corresponding rank variable r_X is defined as the ordinal rank for variable X. Given two variables X and Y and a data set of n samples, the difference between the rankings is computed from the rank variables as $d_i = (r_X)_i - (r_Y)_i$ for each sample $i = 1, ..., n$. Spearman's rank correlation coefficient between variables X and Y is then defined as

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n^3 - n}. \tag{3.14}$$

As proven by Kendall (1970, page 8), the value of the coefficient is between -1 and $+1$, the boundary value $+1$ signifying perfect positive correlation and value -1 denoting perfect negative correlation between the two ranked variables. Furthermore, as shown by Kendall (1970, page 23), a larger value of the Spearman's rank correlation coefficient signifies stronger positive correlation between the ranked variables. Values close to zero thus signify little to no correlation. Therefore, the Spearman's rank correlation coefficient between the values predicted by the algorithm and the actual target values measures how well the predicted order matches the reality.

In case the data set contains a large number of tied ranks, a correction factor should be used as explained by Siegel (1956, page 206).

The results are presented in Section 4.2.

3.8.2 Drug sensitivity data

The prediction performance of the presented mechanism is evaluated as in the paper Honkela et al. (2017) on the drug sensitivity data and compared with the aforementioned (see Section 3.8.1) two non-private versions of itself as well as with three methods using fixed precision parameters: private linear regression with no clipping, output perturbed

linear regression, and functional mechanism linear regression. Output perturbed linear regression (Wu et al., 2015) perturbs the regression parameters instead of the sufficient statistics, and functional mechanism linear regression (Zhang et al., 2012) is an objective perturbation method.

The prediction performance of each method is evaluated in test settings where the reduced data dimensionality is set as $d = 10$, the number of available non-private data is set as $n_{npv} = 10$ and the size of the private data set is varied in $n_{pv} = 0, 100, \ldots, 800$. In each case, two privacy budget values $\epsilon = 1$ and $\epsilon = 2$ are tested. The prediction performance with fixed precision parameter values and 10 non-private 10-dimensional data points is used as a baseline, and as an additional baseline, the corresponding result with 64-dimensional data is presented.

The trade-offs of the differentially private learning are also tested with robust private linear regression in three aspects: the number of available private data ($n_{pv} = 0, 100, \ldots, 800$) versus a) the data dimensionality ($d = 5, 10, \ldots, 40$), b) the number of additional non-private data ($n_{npv} = 0, 5, \ldots, 30$), and c) the privacy budget ($\epsilon = 1, 1.5, \ldots, 3$). The best prediction accuracy with only 10 non-private data points is used as the baseline, and the prediction performance is reported as the relative improvement over the baseline.

In each test setting and for each of the 265 drugs, the GDSC data set is split into training and test sets similarly as with the synthetic data in a 50-fold Monte Carlo cross-validation. In each split, the validation set consists of the first $n_{test} = 100$ samples and the rest are split into private and non-private training sets of determined sizes. After the splitting, the cell lines with missing responses to the selected drug are discarded. The model is then trained on the training data and the fitted model is used to compute the prediction on the validation data as in (3.5) for models using fixed precision parameters and as in (3.7) for models assigning priors for the precision parameters. The accuracy of the prediction is evaluated, and the results are averaged over all drugs and cross-validation folds.

The accuracy of the prediction is computed as the Spearman's rank correlation coefficient (3.14) between the actual drug responses of the cell lines and the corresponding predicted drug responses of the cell lines for the validation set. Larger values imply better accuracy: the predicted cell line rankings according to the sensitivity to a certain drug match the reality more accurately. Values close to zero or negative values mean the prediction is rather poor and the rankings do not match. Spearman's rank correlation is a reasonable choice for the performance measure since in a real-life application, the exact value of the drug response is not as relevant as knowing which drugs best work on a certain cancer in order of effectiveness and which have no effect.

The results are presented in Section 4.3.

3.9 Implementation

The algorithm and tests are implemented in Python, and the source code is available at `https://github.com/DPBayes/robust-private-lr`. Since some of the tests require too extensive resources to be run on a regular desktop computer in any sensible time, they are instead run on the Science-IT project's Triton computer cluster at Aalto University.

The model (3.1), (3.6) is build and inference carried out with the PyMC3 Python module (Salvatier et al., 2016). The hyperparameters for the gamma priors (3.6) of the precision parameters λ, λ_0 are set to $a = b = a_0 = b_0 = 2$, as the Gamma$(2, 2)$ distribution has mean 1 and variance 1/2 and thus defines a realistic distribution over sensible values of the precision parameters which should be larger than zero. The model is learnt using PyMC3's ADVI which fits a normal distribution with uncorrelated variables to the posterior. The fitted variational distribution is then sampled and the predictions are computed as in (3.7). ADVI is chosen over MCMC sampling as in this application they produce similar results in this application but ADVI is a significantly faster alternative.

As explained in Section 3.4.1, the optimal privacy budget split is decided based on prediction performance averaged over five synthetic 10-dimensional auxiliary data sets of a size that is approximately half of the actual data set size (500 samples), and over five noise samples assuming $\epsilon = 2$, and for each split, the optimal clipping thresholds are chosen similarly based on average prediction performance over five auxiliary data sets and five noise samples. In order to evaluate the accuracy with each split, a variational distribution is fitted to the posterior using ADVI and the model prediction is computed as in (3.7) based on $m = 5000$ samples drawn from the variational distribution. The final optimal projection thresholds for each test setting with different number of available private samples, data dimensionality, and total privacy budget, are chosen using the optimal budget split and based on prediction performance averaged over 20 synthetic auxiliary data sets and 20 noise samples. All auxiliary data are generated with the precision parameter values fixed to their prior means, $\lambda = \lambda_0 = 1$. The predictions with each pair of clipping thresholds are also computed as in (3.5) using these fixed values for the precision parameters, as using the sampling method with ADVI for all test cases would be infeasible in practice.

The two modified, non-private versions of the robust private linear regression are implemented similarly as the original algorithm except that they do not add noise to the sufficient statistics and one of them does not apply data projection. The three compared methods (private linear regression, output perturbed linear regression, and functional mechanism linear regression) were implemented in Matlab by M. Das. The implementation of the output perturbed linear regression deploys the minConf optimisation package (Schmidt et al., 2009), and the functional mechanism version uses the code provided at `https://sourceforge.net/projects/functionalmecha`.

Chapter 4

Experimental results

4.1 Privacy budget split

Different privacy budget splits were studied on a synthetic 10-dimensional data set consisting of 500 samples, and the total privacy budget was assumed to be $\epsilon = 2$. In these tests, the optimal split turned out to be $p_1 = 0.35, p_2 = 0.6, p_3 = 0.05$, which means the largest portion, 60% of the budget is assigned to the term $X^T y$, the second largest 35% portion to the term $X^T X$, and the remaining 5% to the term $y^T y$. As can be seen in Figure 4.1, the accuracy of the algorithm improves when a smaller share is given to the term $y^T y$. When the share for $y^T y$ is kept constant, prediction accuracy seems to peak at a point where the budget share for the term $X^T y$ is larger than the share for the term $X^T X$. This indicates the term $X^T y$ should be prioritised and given a larger share of the privacy budget than the others — it might be more important in the model fitting or more sensitive to noise. Intuitively, this makes sense since $X^T y$ is the only term that ties the explanatory data X and dependent data y together. The second most important term seems to be the $d \times d$ matrix $X^T X$, and the least important term is the scalar $y^T y$, which seems rather plausible considering the number of elements in each term.

4.2 Synthetic data

The green curve in Figure 4.2 shows the prediction accuracy of the algorithm on 10-dimensional synthetic data of 500 private and 10 public samples with different values of the privacy parameter ϵ. The dashed purple curve represents the corresponding non-private algorithm with clipping and the dashed orange curve denotes the non-private algorithm with no clipping. The prediction performance of the non-private non-clipping version is independent of the privacy parameter and thus stays constant. The prediction

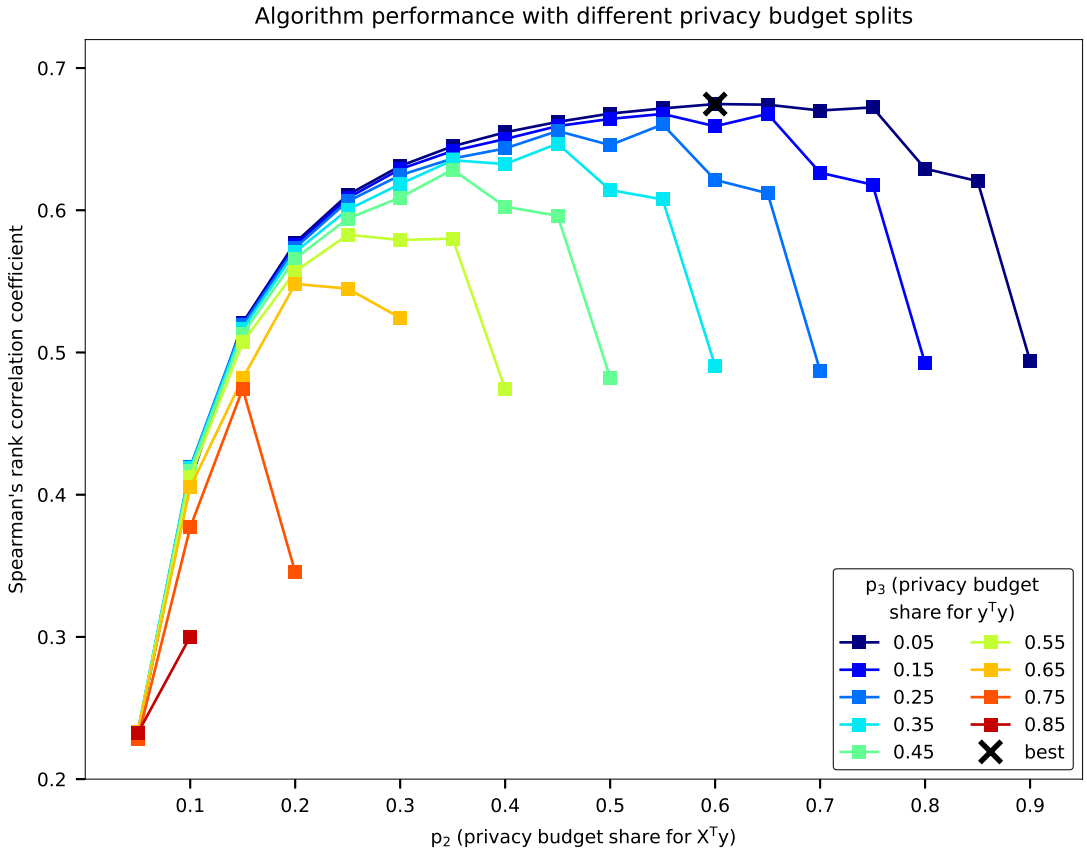

Figure 4.1: **The prediction accuracy of the algorithm on synthetic auxiliary data with different privacy budget splits.** The values on the x-axis indicate how large proportion of the budget is assigned to the term $X^T y$, coloured lines denote different budget shares for the term $y^T y$, and the remaining portion of the privacy budget is left for the term $X^T X$. For clarity, only half of the studied splits are plotted, including the optimal one marked with an 'X': $p_1 = 0.35, p_2 = 0.6, p_3 = 0.05$.

accuracy of the non-private clipping version is almost the same as the former's but there are slight fluctuations due to different clipping thresholds that vary along with the privacy parameter. With larger values of the privacy parameter, the prediction performance of the robust private linear regression algorithm is similar to the performance of the non-private versions. Smaller values of the privacy parameter guarantee stronger privacy but the algorithm accuracy deteriorates: as the accuracy of the sufficient statistics decreases, the results of the model sampling also get less accurate. Moreover, the error bars indicate that the accuracy of the predictions varies a lot when stricter privacy is required. The results are as expected and demonstrate the inevitable trade-off between privacy and accuracy.

The results in Figure 4.3 are from the same setting except that now the privacy parameter is kept constant $\epsilon = 2$ and the number of available data is varied instead. The non-private versions are rather similar in performance: first, the prediction performance quickly improves when the size of the data set increases, then it remains stable. The prediction accuracy of the robust private linear regression algorithm first drops a bit when the first 100 private samples are added to the data set but then quickly recovers and constantly improves, nearly reaching the accuracy level of the non-private methods when the size of the data set has reached 10 public and 800 private samples. The initial drop is observed because the prediction performance is better with a small data set and accurate statistics than with a slightly larger data set and perturbed statistics: the added noise hinders the prediction accuracy more than the amount of added data improves it. However, the experienced drawback is quickly overcome when the size of the private data set further increases. Moreover, the prediction accuracy varies a lot when only a small data set is available, but as the size of the data set increases, the observed variation decreases and the algorithm constantly outputs good results.

4.3 Drug sensitivity data

Figure 4.4 displays the prediction performance of the robust private linear regression algorithm on the GDSC drug sensitivity data with three colour map plots that demonstrate the trade-offs of differential privacy. The prediction accuracy is presented as relative improvement over the baseline result (10 non-private 40-dimensional data points). Plot a) shows that as the dimensionality of the data increases, more data are needed to reach good accuracy. The dimensionality d directly affects the amount of added noise in Algorithm 2: higher dimensionality d corresponds to a wider noise scale, and consequently, the perturbed statistics are less accurate. The genes are ordered in descending order according to the importance estimated based on the number of mutation counts, and dimensionality d means that the d most important genes are taken in the data set. The rippling in higher dimensionalities is probably due to the fact that the relevance order

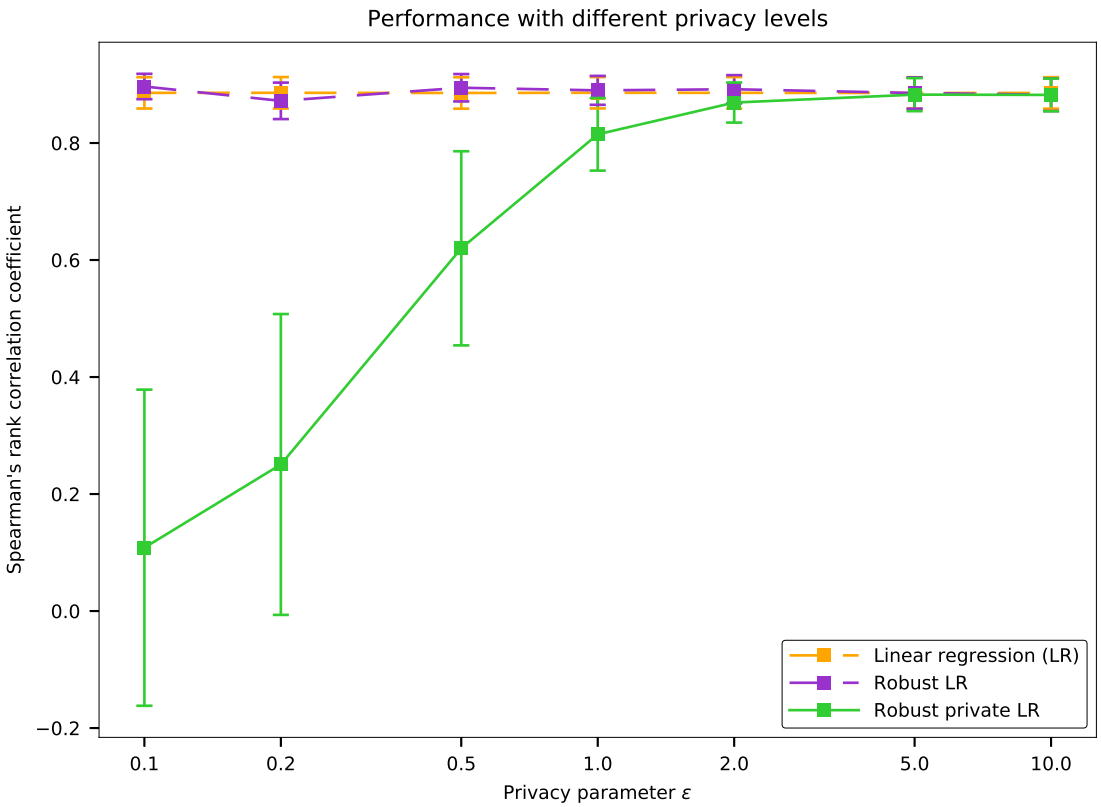

Figure 4.2: **The prediction accuracy of the algorithm on synthetic data as a function of the privacy parameter.** The solid curve represents the robust private linear regression algorithm and the two dashed curves denote the non-private versions. The square markers denote the average results over 50-fold cross-validation and the error bars indicate the corresponding standard deviations. A logarithmic scale is used on the x-axis.

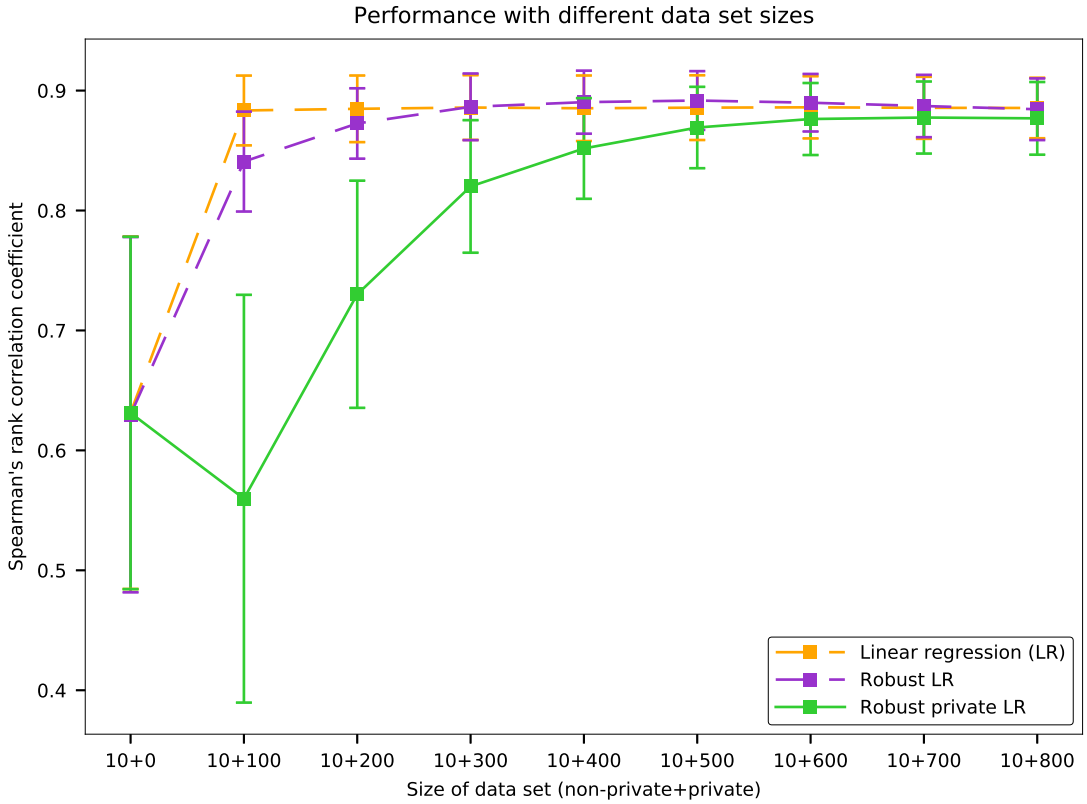

Figure 4.3: **The prediction accuracy of the algorithm on synthetic data as a function of the available data set size.** The solid curve represents the robust private linear regression algorithm and the two dashed curves denote the non-private versions. The square markers denote the average results over 50-fold cross-validation and the error bars indicate the corresponding standard deviations.

of the genes is not perfect. In reality, how much each gene affects the development of cancer is likely a highly complex question and the number of mutations does not explain everything. Furthermore, the high amount of noise can potentially affect the goodness of the chosen clipping thresholds, which may affect the observed behaviour. It is also possible some other budget split would be better at higher dimensionalities. Subplot b) shows that the size of the small additional public data set only matters if there are very little private data available. Plot c) again demonstrates how stricter privacy guarantees require more data for the algorithm to reach good accuracy.

The prediction accuracy is plotted as a function of the data set size in Figure 4.5 (with fixed privacy parameter $\epsilon = 2$). It is compared against the prediction performance of the two non-private versions of the same algorithm and three different private methods. The baselines show the prediction accuracy of the non-private non-clipping version using fixed precision parameter values and only 10 public data points of either 10-dimensional or 64-dimensional data. The prediction performance of the non-private versions quickly improves when the size of the data set increases. Clipping slightly worsens the accuracy in non-private learning. The accuracy of the robust private linear regression algorithm also rapidly increases as the amount of available data grows, nearly reaching the accuracy level of the non-private versions. Meanwhile, the other private methods perform relatively poorly: The accuracy of the output perturbed version improves very slowly as the size of the data set increases and it never reaches the baseline result with the studied data set sizes. The accuracy of functional mechanism and private linear regression methods stays at nearly zero and fails to significantly improve even though the amount of available data increases. Out of the private methods, the robust private linear regression performs the best, being the only one that remarkably benefits from the growing amount of data and reduces the gap to the non-private algorithm. The sampling methods produce larger error bars since the accuracy of the results varies more across the cross-validation. The robust private linear regression method exhibits behaviour observed also with synthetic data: as the size of the data set is small, the accuracy of the results varies a lot, but as the data set size increases, the variation decreases.

Figure 4.6 shows the corresponding tests with stricter privacy ($\epsilon = 1$). The private methods naturally experience a decrease in accuracy but otherwise perform in a similar manner as in the tests in Figure 4.5. The observed drop in the accuracy of the robust private linear regression mechanism at 10 non-private and 100 private data samples is deeper than with more loose privacy, but the prediction performance again quickly recovers as more private data are added.

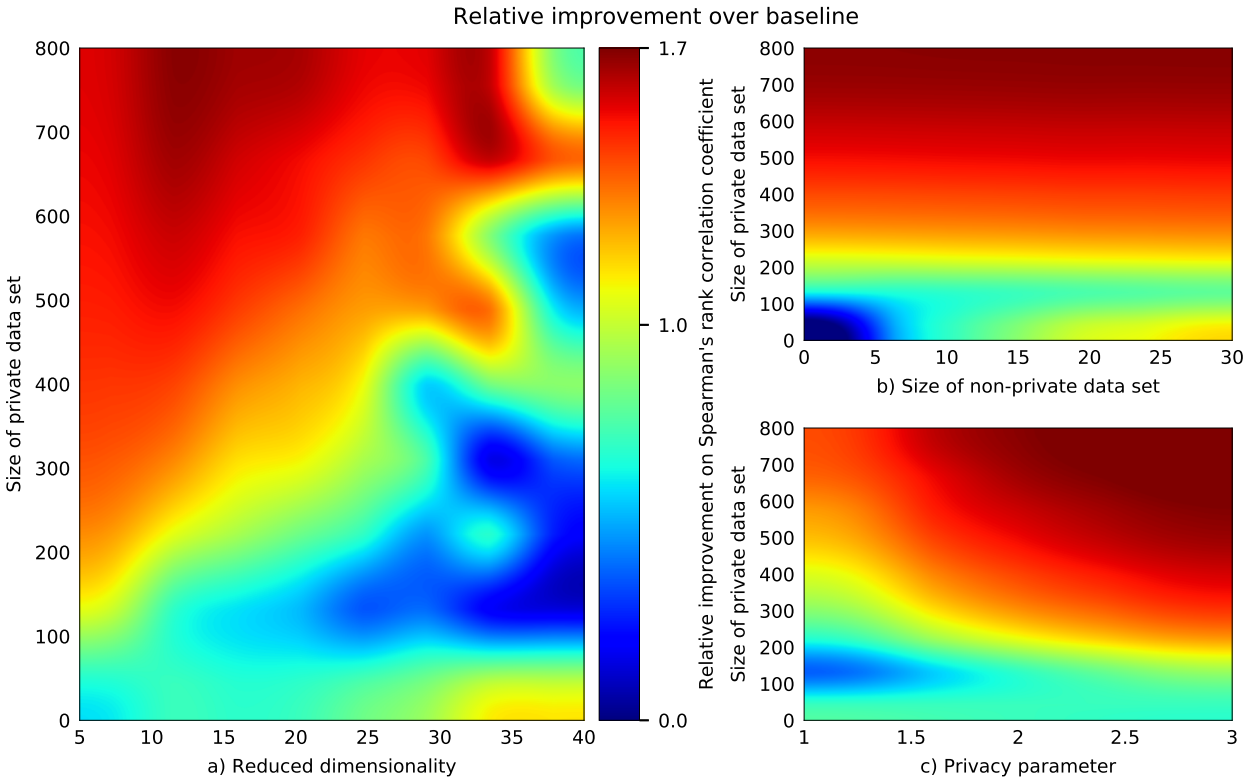

Figure 4.4: **The trade-offs of differential privacy.** The prediction accuracy of the robust private linear regression algorithm on the GDSC data is presented as the relative improvement over the baseline result that uses 10 non-private 40-dimensional data points. All results are averages over 50-fold cross-validation. If not otherwise specified, the test cases use 10-dimensional data, 10 additional non-private data points, and privacy budget $\epsilon = 2$.

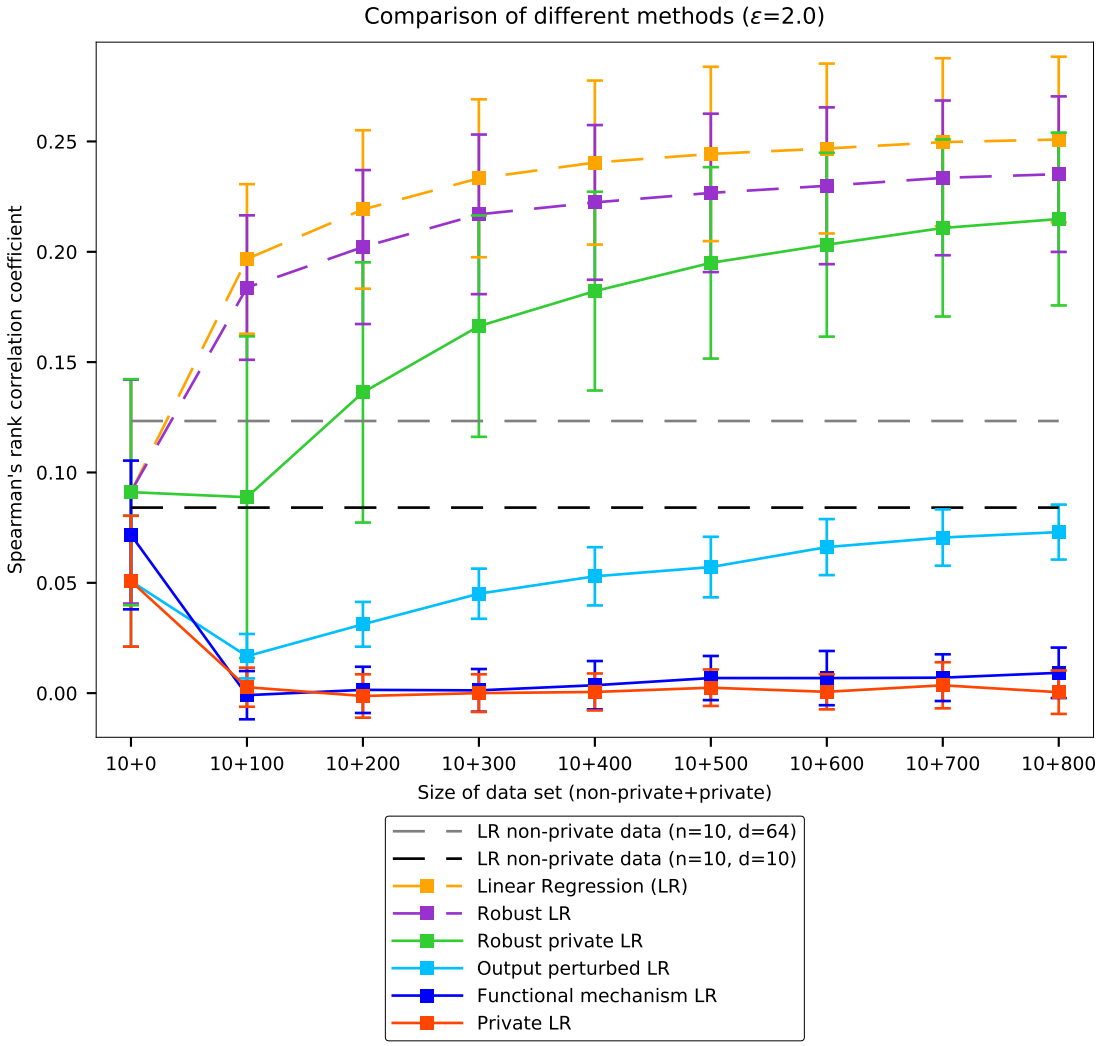

Figure 4.5: **The prediction accuracy of the algorithm on GDSC data as a function of the available data set size.** The black dashed line represents the baseline result computed using fixed precision parameter values and only 10 non-private 10-dimensional data points with no private data. The grey dashed line is the corresponding result with 64-dimensional data. The other two dashed curves denote the non-private versions of the robust private linear regression, and solid lines represent the compared differentially private methods. All methods use 10-dimensional data unless otherwise specified. The square markers denote the average results over 50-fold cross-validation and the error bars indicate the corresponding standard deviations.

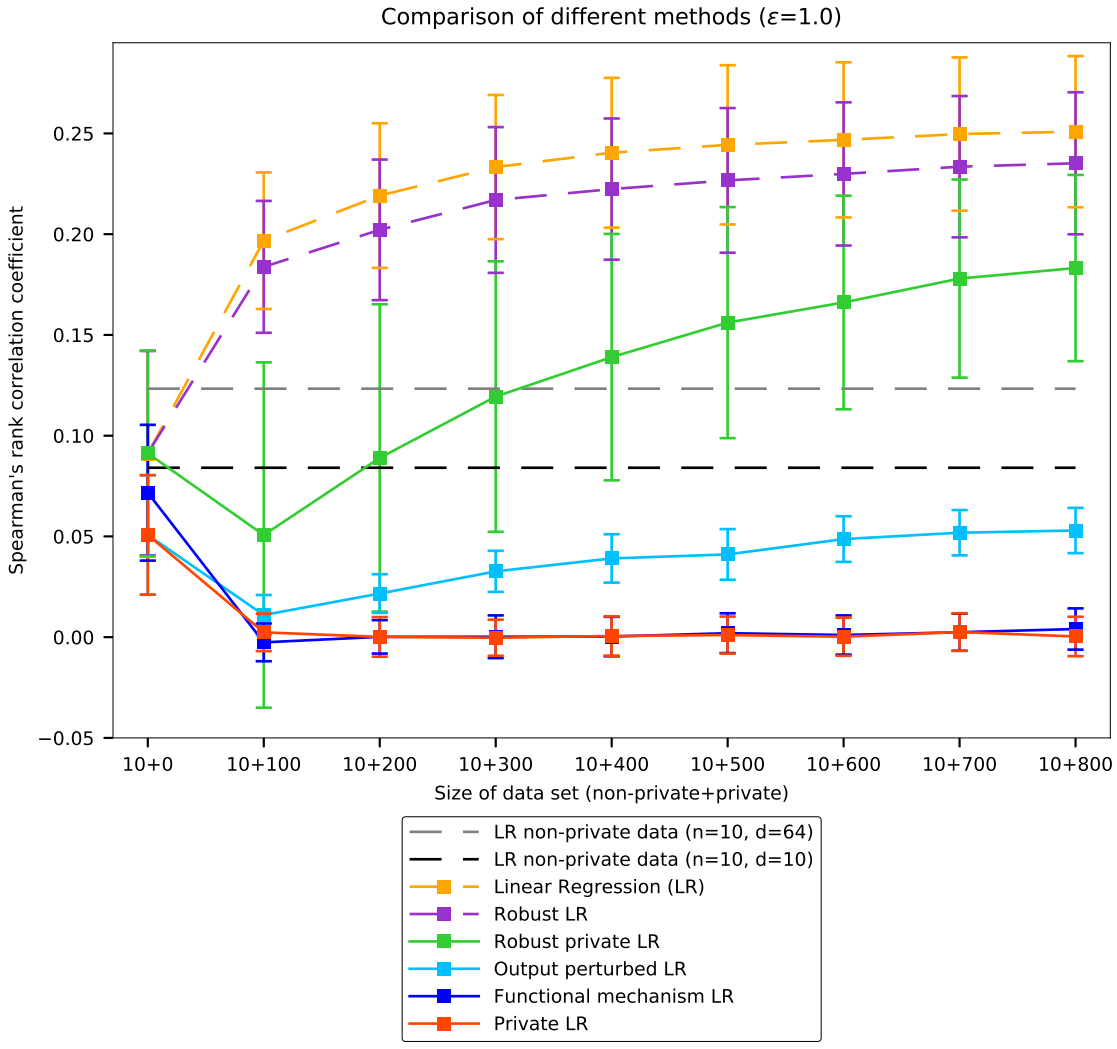

Figure 4.6: **The prediction accuracy of the algorithm on GDSC data as a function of the available data set size.** The setting is otherwise identical to Figure 4.5, but private methods use stricter privacy budget $\epsilon = 1$ instead of $\epsilon = 2$.

Chapter 5

Discussion

5.1 Conclusions

To sum up, the robust private linear regression algorithm behaves as expected on both synthetic and drug sensitivity data. Its accuracy improves with looser privacy and worsens with stricter privacy. With stricter privacy levels, the prediction accuracy can be significantly improved by using more data. The mechanism is able to nearly reach the accuracy of the non-private methods even with moderate sized data sets. The performed tests indicate the algorithm could be used in real-life applications, providing adequate privacy while also allowing accurate data analysis on reasonably sized data sets. The projection of the data inside chosen thresholds is the key factor in the success of the method and without it the algorithm would perform poorly as noted by Honkela et al. (2016). The other compared differentially private methods lose to the proposed algorithm, and they also would not benefit from clipping since they are based on totally different mechanisms.

The methods used to choose the parameters of the algorithm seem to work: The chosen budget split is intuitively sensible and performs well on both synthetic and real-life data. The chosen projection thresholds seem to be well-adjusted to each test case — except maybe at high dimensionalities. Assigning priors to the precision parameters of the model and sampling the fitted variational posterior seem to work as they should and produce good results. Its benefit in accuracy over the fixed precision method is likely moderate (at 10 non-private data points in Figures 4.5 and 4.6 the sampling methods perform slightly better than the baseline) — but still real, and using priors instead of arbitrary precision parameter values makes the analysis more justifiable.

5.2 Own contribution

My contribution to the development of the robust private linear regression algorithm was to implement a new version of the mechanism with several improvements over the old implementation (Honkela et al., 2016): the precision parameters λ and λ_0 of the Bayesian linear regression model (3.1) are now assigned prior distributions instead of fixed values as explained in Section 3.2.1, the privacy budget split (introduced in Section 3.3.1) is now optimised based on prediction performance on auxiliary data as described in Section 3.4.1 instead of using an even split by default, and instead of computing the posterior mean analytically (Equation 3.5), the model prediction is now computed either by MCMC methods or by ADVI as explained in Section 3.2.1. These changes are designed to make the algorithm more robust as they eliminate the need to guess suitable values for the precision parameters and allow utilising the available privacy budget more efficiently. I also updated the related mathematical details of the mechanism (the formal definition of Algorithm 2, the proof of Theorem 3.9, and the log-likelihood 3.8 needed in the implementation of the MCMC and ADVI sampling) as represented in Section 3.3.3, and was responsible for carrying out the new experiments with the new, larger version of the drug sensitivity data set, the results of which are represented in Chapter 4 and in the paper Honkela et al. (2017).

5.3 Future work

In the future, the algorithm could be improved in several ways. For one thing, since perturbed and non-perturbed sufficient statistics have different accuracy, it would make sense to use separate precision parameters (λ) for private and public data. The expanded model could then make better use of the accurate statistics computed from additional public data. It would also be useful to have an analytical justification for using a certain privacy budget split instead of just studying it experimentally — in addition to being more precise and certain, it could bring more insight into the relationships between the used statistics and how noise affects the accuracy of the fitted model. The optimal split may be dependent on many factors, such as the total privacy budget and the dimensionality and the number of available data.

Moreover, the outlier projection idea could be deployed in other kinds of statistical models that could be more complex and thus able to grasp the more delicate properties of the data. Other, more sophisticated dimensionality reduction methods could also be utilised, depending on the application area. One standard example of such a method is principal component analysis (Jolliffe, 2002).

Bibliography

Alwi, Z. B. (2005). The Use of SNPs in Pharmacogenomics Studies. *The Malaysian Journal of Medical Sciences*, 12(2):4.

Baldi, P. and Hatfield, G. W. (2002). *DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling*. Cambridge University Press.

Consumer Financial Protection Bureau (2017). What is a credit report? [Online; retrieved 26th October 2017].

Dwork, C. (2006). Differential Privacy. In *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II*, ICALP'06, pages 1–12, Berlin, Heidelberg. Springer-Verlag.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating Noise to Sensitivity in Private Data Analysis. In Halevi, S. and Rabin, T., editors, *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings*, pages 265–284. Springer Berlin Heidelberg.

Dwork, C. and Roth, A. (2014). The Algorithmic Foundations of Differential Privacy. *Foundations and Trends®in Theoretical Computer Science*, 9(3–4):211–407.

Dwork, C. and Smith, A. (2009). Differential Privacy for Statistics: What we Know and What we Want to Learn. *Journal of Privacy and Confidentiality*, 1(2):135–154.

Gamerman, D. and Lopes, H. F. (2006). *Markov chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Texts in Statistical Science Series. Chapman & Hall/CRC, second edition.

Garnett, M. J., Edelman, E. J., Heidorn, S. J., Greenman, C. D., Dastur, A., Lau, K. W., Greninger, P., Thompson, I. R., Luo, X., Soares, J., et al. (2012). Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature*, 483(7391):570–575.

Heikkilä, M., Lagerspetz, E., Kaski, S., Shimizu, K., Tarkoma, S., and Honkela, A. (2017). Differentially Private Bayesian Learning on Distributed Data. *arXiv e-prints*, 1703.01106v2 [stat.ML].

Honkela, A., Das, M., Dikmen, O., and Kaski, S. (2016). Efficient differentially private learning improves drug sensitivity prediction. *arXiv e-prints*, 1606.02109v1 [stat.ML].

Honkela, A., Das, M., Nieminen, A., Dikmen, O., and Kaski, S. (2017). Efficient differentially private learning improves drug sensitivity prediction. *arXiv e-prints*, 1606.02109v2 [stat.ML].

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer-Verlag GmbH, second edition.

Kendall, M. G. (1970). *Rank correlation methods*. Charles Griffin & Company Limited, fourth edition.

Kifer, D. and Machanavajjhala, A. (2011). No Free Lunch in Data Privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 193–204, New York, NY, USA. ACM.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic Differentiation Variational Inference. *Journal of Machine Learning Research*, 18(14):1–45.

McSherry, F. and Talwar, K. (2007). Mechanism Design via Differential Privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103. IEEE.

Narayanan, A. and Shmatikov, V. (2007). How To Break Anonymity of the Netflix Prize Dataset. *arXiv e-prints*, cs/0610105v2 [cs.CR].

Naveed, M., Ayday, E., Clayton, E. W., Fellay, J., Gunter, C. A., Hubaux, J.-P., Malin, B. A., and Wang, X. F. (2015). Privacy in the Genomic Era. *arXiv e-prints*, 1405.1891v3 [cs.CR].

Nuremberg Military Tribunals (1949). The Nuremberg Code. *Trials of War Criminals before the Nuremberg Military Tribunals under Control Council Law*, 2(10):181–182.

Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2:e55.

Sarwate, A. D. and Chaudhuri, K. (2013). Signal Processing and Machine Learning with Differential Privacy: Algorithms and Challenges for Continuous Data. *IEEE signal processing magazine*, 30(5):86–94.

Schmarzo, W. D. (2013). *Big Data: Understanding how data powers big business.* John Wiley & Sons.

Schmidt, M., Berg, E., Friedlander, M., and Murphy, K. (2009). Optimizing Costly Functions with Simple Constraints: A Limited-Memory Projected Quasi-Newton Algorithm. In van Dyk, D. and Welling, M., editors, *Proceedings of The Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS) 2009*, volume 5, pages 456–463, Clearwater Beach, Florida.

Sears, B. and Mallory, C. (2011). Documented Evidence of Employment Discrimination & Its Effects on LGBT People. Technical report, The Williams Institute.

Siegel, S. (1956). *Nonparametric Statistics for the Behavioral Sciences.* McGraw-Hill Book Company, INC., international student edition edition.

Singel, R. (2009). Netflix Spilled Your Brokeback Mountain Secret, Lawsuit Claims. [Online; retrieved 5th October 2017].

Singel, R. (2010). NetFlix Cancels Recommendation Contest After Privacy Lawsuit. [Online; retrieved 5th October 2017].

Sweeney, L. (2000). Uniqueness of Simple Demographics in the U.S. Population, LIDAPWP4. Technical report, Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA.

Sweeney, L. (2002). K-anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570.

TNS Opinion & Social (2015a). Special Eurobarometer 431 : Data protection. Technical report, European Commission Directorate-General for Communication.

TNS Opinion & Social (2015b). Special Eurobarometer 437: Discrimination in the EU in 2015. Technical report, European Commission Directorate-General for Communication.

Wu, X., Fredrikson, M., Wu, W., Jha, S., and Naughton, J. F. (2015). Revisiting Differentially Private Regression: Lessons From Learning Theory and their Consequences. *arXiv e-prints*, 1512.06388 [cs.CR].

Yang, W., Soares, J., Greninger, P., Edelman, E. J., Lightfoot, H., Forbes, S., Bindal, N., Beare, D., Smith, J. A., Thompson, I. R., et al. (2013). Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Research*, 41(Database issue)(D1):D955–D961.

Zhang, J., Zhang, Z., Xiao, X., Yang, Y., and Winslett, M. (2012). Functional Mechanism: Regression Analysis under Differential Privacy . *Proceedings of the VLDB Endowment*, 5(11):1364–1375.