Contributions to Probabilistic and Statistical Foundations of Differential Privacy

by

Devyani Biswal

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Ph.D. degree in
Mathematics and Statistics[1]

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

[1]The Ph.D. program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics.

Abstract

It is undeniable that the rapid advancement of data analytics and artificial intelligence over the past decade has transformed many industries. However, these advancements have also highlighted the need for robust privacy-preserving techniques to protect personal data from misuse. Furthermore, increasing regulatory scrutiny and public awareness highlight the importance of protecting individual privacy as data-driven technologies evolve. This thesis addresses these concerns by exploring and advancing the field of differential privacy. The primary goal of this thesis is to provide a mathematical and statistical framework for differential privacy to better balance and answer questions related to data utility and privacy. Indeed, the majority of research up to date focuses on privacy aspects, with little emphasis on data utility. As such, the thesis investigates privacy guarantees across different settings and statistical problems. We propose many novel mechanisms that integrate concepts from statistical disclosure control, statistics, time series, and machine learning along with classical differential privacy. For this we explore various extensions of ε- and (ε, δ)-differential privacy mechanisms. We prove the validity (from both the privacy and data utility perspective) of these proposed mechanisms using a rigorous mathematical framework. The theoretical results are complemented by a variety of numerical experiments to validate the underlying intuitions. The findings indicate that our contributions significantly improve data utility while offering strong privacy guarantees. As such, they can be practically implemented in the real-world settings.

Acknowledgements

I wish I could adequately express my gratitude to my supervisor, Dr. Rafal Kulik. Thank you for allowing me to pursue my goals not only in academia but also in sport and life. Very few people are fortunate enough to have a supervisor who supports all these endeavours at once, and for that, I am deeply thankful.

I must also express my immense appreciation to Luk Arbuckle for supporting me as an industry supervisor and introducing me to the world of data privacy. Your guidance has been invaluable, and without your support, I would not be where I am today.

My sincerest thank you to my defence committee for generously sharing your knowledge and expertise.

My work would not be possible without the funding and support of the MITACS Accelerate Fellowship, which supported me for 3 years of my research, and Privacy Analytics for providing me a place to pursue my research and give it a real-world purpose.

To my parents, thank you for being exemplars of life in academia. Growing up in universities opened my eyes to academia and significantly influenced my pursuit of this path. I am so thankful to have parents that helped me edit, think critically, and never to give up.

To my super smart, supermodel, twin sister - Divya. Thank you for being the Mary-Kate to my Ashley.

And finally, to my partner in life, Rostam, thank you for always being there for me. Without your encouragement and belief in me I am certain I wouldn't be writing this page today.

The best way to guard privacy? Pretend you don't exist!

Contents

Abstract ii

Acknowledgements iii

List of Tables xi

List of Figures xiii

1 Introduction 1
- 1.1 Preface . . . 1
- 1.2 Introduction . . . 2
- 1.3 Structure of the thesis . . . 5
- 1.4 Thesis Contributions . . . 7

2 Mathematical foundations 11
- 2.1 Notation and basic terminology . . . 11
- 2.2 Probability distributions . . . 17
- 2.3 Distance between probability distributions . . . 18

3 Techniques for disclosure control 21
- 3.1 k-anonymity . . . 21
- 3.2 PRAM . . . 23
- 3.3 k-PRAM . . . 24
- 3.4 Noise addition . . . 25
- 3.5 k-noise . . . 25
- 3.6 Experimental Results . . . 28
- 3.7 Conclusion . . . 29
- 3.8 Figures . . . 30

4 Differential privacy 35
- 4.1 Introduction . . . 35
- 4.2 Basic definition . . . 36
- 4.3 Laplace noise and differential privacy . . . 38

4.4 (ε, δ)-Differential Privacy . . . 39
4.5 Properties . . . 41
4.5.1 Preservation of differential privacy under different queries . . . 41
4.5.2 Post-processing . . . 43
4.5.3 Group privacy . . . 48
4.5.4 Compositions . . . 49

5 Differential Privacy from a data utility perspective 51
5.1 Introduction . . . 51
5.2 Dealing with sensitivity . . . 54
5.2.1 Smooth sensitivity . . . 55
5.2.2 Towards general sensitivity . . . 63
5.3 Mixed Noise Mechanism (MNM) . . . 73
5.4 Blocking . . . 92
5.4.1 Algorithm Block-DP I . . . 92
5.4.2 Algorithm Block-DP II . . . 96
5.5 Bounded Laplace Mechanism . . . 98
5.6 Pre-processing vs Post-processing . . . 100
5.7 Confidence Intervals . . . 108
5.8 Changing the distance between probability distributions . . . 111
5.9 Conclusion . . . 115

6 Time series 117
6.1 Introduction . . . 117
6.2 Differentially private queries in times series . . . 118
6.2.1 Data release and attack scenarios . . . 119
6.3 Privacy leakage for time series . . . 120
6.3.1 Total dependence and independence . . . 120
6.3.2 Privacy leakage for the mean . . . 121
6.4 Conclusion . . . 138

7 Towards Machine Learning and Differential Privacy 141
7.1 Introduction . . . 141
7.2 Preliminaries . . . 142
7.3 Gradient Descent . . . 144
7.4 Stochastic optimization problem . . . 145
7.5 Stochastic Gradient Descent (SGD) . . . 147
7.6 Differentially Private Stochastic Gradient Descent . . . 148
7.7 Convergence of the algorithm . . . 151
7.8 Comments . . . 153
7.9 Appendix: Computations for conditional expectations . . . 154

8 Conclusion and future direction of research **159**
8.1 Conclusion 159
8.2 Future work 160

9 Bibliography **163**

Index **167**

Appendices **169**

A R codes **171**

List of Tables

3.1 k-anonymity example 22
3.2 3-anonymity for 1 variable 23
3.3 3-anonymity for 2 variables 23
3.4 Utility results for k-PRAM & k-noise 29

5.1 MNM confidence intervals for various distributions 90
5.2 Block DP-I vs Block DP-II 98

List of Figures

2.1 Drawing: sanitized response mechanism 15
2.2 Drawing: output perturbation mechanism 16

3.1 k-noise group 26
3.2 k-PRAM 30
3.3 k-noise 31
3.4 k-PRAM vs k-noise 32
3.5 Expected group size 33

4.1 Histogram of Laplace-Normal density 47

5.1 Weighted sensitivity - uniform 70
5.2 Weighted sensitivity - truncated exponential 71
5.3 δ-privacy 72
5.4 δ values for ADP 75
5.5 Drawing: MNM 77
5.6 MNM results - normal distribution 82
5.7 MNM - Box plot for the normal 83
5.8 MNM results - student-t distribution 84
5.9 MNM - Box plot for the student-t 85
5.10 MNM results - exponential distribution 86
5.11 MNM - Box plot for the exponential 87
5.12 MNM results - Pareto distribution 88
5.13 MNM - Box plot for the Pareto 89
5.14 MNM results for the median 91
5.15 Post-processing vs Pre-processing for the median 107

6.1 DP in time series: A1+N3 scenario 127
6.2 DP in time series: A2+N1 scenario; first example 132
6.3 DP in time series: A2+N1 scenario; second example 133
6.4 DP in time series: A2+N1 scenario; third example 134
6.5 DP in time series: A2+N1 scenario; fourth example 135

Chapter 1

Introduction

1.1 Preface

In the recent years, the rapid advancements in data and artificial intelligence (AI) have significantly transformed a number of industries, including finance and healthcare. These technological advancements have enhanced the ability to derive insights from vast quantities of data, driving innovation and improving decision-making processes. However, this exponential growth does not come without concerns, namely the individual's right to privacy. As organizations increasingly rely on personal data to do statistical analysis, to train machine learning models or combine data in ways that have not been done before, the risk of compromising individual privacy becomes a key consideration. The occurrence of high-profile data breaches and the misuse of personal data has led to heightened public awareness and regulatory scrutiny. Consequently, the necessity for robust yet practical privacy-preserving techniques has become apparent. Legislation such as the General Data Protection Regulation (GDPR) in Europe and Bill C-27 in Canada reflects these heightened privacy concerns and the necessity for robust data protection frameworks.

This context provides the motivation to study privacy-preserving techniques that have emerged as key techniques for mitigating these concerns while still producing useful data. The most popular and promising technique from a data utility perspective is differential privacy, which originates from computer science. A principal challenge is to formulate what data privacy means from a mathematical and statistical perspective, and then to determine how differential privacy affects data privacy and data utility. Furthermore, it is essential to comprehend how technical privacy models can facilitate the process of rendering data non-identifiable, and, more crucially, enabling its use for numerous initiatives that contribute to enhanced societal outcomes.

The primary objective of this thesis is to establish a foundational framework within which mathematical and statistical theory can be studied. This framework aims to unify the language and frameworks of statistical theory with those of computer science.

Once this foundation was established, we are to clarify the existing results and make their presentation suitable for mathematical and statistical audience. Furthermore, we propose novel mechanisms and techniques for the practical implementation of differential privacy. Moreover, we investigate more intricate issues, such as the impact of temporal dependence in data and the potential implications for privacy. This seems elementary, but in fact is not trivial.

1.2 Introduction

In the modern era, data is everywhere. It has permeated every aspect of our world. This is not just a byproduct of technological advancement. It is a fundamental shift in how we interact with the world. The process of data collection has become more sophisticated than ever before. It enables not only the aggregation of vast amounts of information but also its transformation into actionable insights. These capabilities are essential for tackling complex societal challenges, whether it's optimizing resource allocation or improving public health outcomes. It is clear that the intentional and ethical use of data can significantly elevate the efficacy of decision-making processes, thereby fostering societal advancement and ensuring more equitable outcomes across diverse communities.

Moreover, data is now a significant economic commodity, influencing markets and policy decisions on a global scale. Its value extends beyond numerical input; it captures potential insights into economic trends, predictive analytics, thus driving innovation and strategic planning. The commodification of data makes it clear that it has two roles: that of a resource and that of a product. This means that we must think carefully about how it is collected, used and shared. The transition of data into an economic asset demands robust data governance frameworks that not only protect individual privacy but also ensure the fair and responsible use of data.

As data assumes a central role in societal functions, the issue of privacy emerges as a paramount concern. The right to privacy of data is a fundamental human right, recognized and upheld by various legal frameworks around the world, such as the General Data Protection Regulation (GDPR) in the European Union. Country regulations emphasize the necessity of consent, transparency and the right to erasure, ensuring individuals have control over their personal information. However, the challenges of maintaining data privacy are compounded by the rapid evolution of technology and the sophistication of data collection techniques. These developments necessitate continuous updates to legal protections to safeguard personal data against unauthorized access and misuse, reinforcing the need for practical approaches to privacy.

Privacy laws and regulations are primarily concerned with identifiable information. Although the language varies by law or regulation, the likelihood of identifying a data

subject is commonly understood as the legal test of whether privacy laws or regulations apply. The General Data Protection Regulation (GDPR) of the European Union states [21] that "to determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling out, either by the controller or by another person, to identify the natural person directly or indirectly." From a technical perspective, interpreting "reasonably likely" is challenging. The term "likely" is difficult to translate into a probability, as demonstrated in [37]. However, it is crucial to understand expectations in privacy laws and regulations to determine when data can be considered non-identifiable and therefore non-personal.

The need to address legal and regulatory expectations through technical methods to satisfy those expectations is not new. Nor it is new to have to ensure that technical models produce useful data in a range of different contexts. This is why there are industry efforts to standardize that guidance through risk-management frameworks for anonymization and in emerging standards, e.g. [30], [31], [25]. Standards ensure current best practices are agreed upon by a wide range of stakeholders, are well documented, and can be used to evaluate implementations to ensure adherence. Regular updates keep standards current while ensuring well-established methods are incorporated. Therefore, the reasonableness with which identifiability will be judged may evolve over time and needs to be monitored for changes. A spectrum of identifiability has even been recognized by industry, incorporating risk-based framing so that a scalable and proportionate approach to compliance is provided. Technical models are incorporated into metrics that attempt to quantify measures of what may constitute a disclosure. These metrics can capture a range of views that may or may not incorporate identifiability.

Opinions may vary regarding what should be considered "anonymized", often focusing on the output of anonymization alone. In the field of statistical disclosure control, however, threat modelling to establish attack scenarios is a well-established practice that is now increasingly common in guidance, [20]. Objective factors are therefore incorporated to determine effective bounds on data recipients or intruders, to determine what may be deemed reasonable. Anonymization is ultimately a form of data protection, and thus can be considered a privacy-enhancing measure. To be effective, privacy models must be used in practice, which means they must also be practical and, when used in the context of anonymization, produce useful data [2].

From a statistical perspective, data privacy is about implementing methodologies that minimize the risk of identifying individual subjects within datasets (micro or macro) while still allowing for the extraction of meaningful insights. It is vital to **achieve this balance between privacy and data utility** in fields such as medical research and finance, where data can often contain sensitive information. Statistical disclosure techniques and anonymization are used to obscure the identity of participants. These statistical approaches are designed to prevent the disclosure of personal attributes while

maintaining the utility of the dataset, thereby safeguarding individual privacy without sacrificing data quality. See [19].

Further complicating the statistical landscape is the issue of data re-identification. This is a process by which anonymized data can be cross-referenced with other public datasets to identify individuals. This highlights a significant vulnerability in traditional statistical disclosure techniques. They are inadequate in the face of sophisticated data mining technologies and the immense amount of data available. See [38].

Statistical disclosure control (SDC) techniques are essential for protecting individual privacy when releasing data for public or research purposes. These techniques employ a range of methods to minimize the risk of identifying individuals in aggregated datasets ([46]). The most widely used SDC methods collectively aim to preserve the statistical properties of the data, such as the mean, variance, and correlations. This ensures that the data remains useful despite any modifications for privacy.

Furthermore, the application of these SDC techniques must be carefully calibrated to balance data utility with privacy protection. Over-application renders the data useless and removes its intended utility. Conversely, under-application leaves individuals exposed to re-identification risks. Therefore, the choice of techniques depends on the specific context of the data and the sensitivity of the information it contains. Ultimately, the implementation of SDC techniques is a decision-making process that requires a deep understanding of both statistical theory and the ethical implications of data release. A lot of SDC methods are relatively standard and are based on simple statistical and probabilistic concepts ([28], [29]).

Differential privacy, [16], emerged as a framework in the beginning of the 21st century, primarily developed by computer scientists to address the growing concerns around data privacy that accompanied the increasing ease of access to detailed data. The concept was formalized by Cynthia Dwork who introduced a cryptographic model designed to ensure that analyses of data (**queries**) do not compromise the privacy of individual data subjects. The principle behind differential privacy is to add a controlled amount of random noise to either the data itself or the outputs (queries) derived from the data, thus obscuring the contributions of individual data points. By doing so, it guarantees that the output of any analysis is less sensitive to any single individuals' data, effectively masking personal information within large datasets. This approach has been particularly influential in fields dealing with sensitive information.

However, the application of differential privacy is not without its challenges and trade-offs. One of the primary advantages of differential privacy is its strong, mathematically proven privacy guarantees, which protect against a wide range of privacy attacks. A challenge is knowing how much noise to add, what kind of noise to add, and how to

measure the effectiveness across different datasets and applications [8]. This challenge only gets exasperated when moving to more complex modelling problems in time series and machine learning applications [9], with a particular focus on differentially private empirical risk minimization, Stochastic Gradient Descent or Coordinate Descent algorithms ([15], [1], [43], [4], [47], [36]).

Balancing privacy protection with data utility is a key challenge in the implementation of differential privacy. The complexity of choosing parameters for the privacy budget has been a barrier to practical implementation. Another challenge is a lack of a rigorous mathematical and statistical framework. Indeed, a bulk of major developments in the field differential privacy stems from the computer science perspective, and often lacks mathematical and statistical rigour. The first serious attempt to formalize differential privacy using the proper statistical language should be attributed to [45]. It allows for proper statistical inference, see [3], and to develop rates of convergence using advanced tools from mathematical statistics ([12]).

In the thesis, we will develop and use the proper mathematical framework that allows for the study of statistical inference and differential privacy in a practical setting. We address both **data privacy** and **data utility** issues.

1.3 Structure of the thesis

In Chapter 2 we collect basic terminology. To be more specific:

- Section 2.1 covers the notation and basic terminology used in the context of probability spaces and metric spaces, providing definitions and examples.
- Section 2.2 introduces probability distributions, focusing on the Gaussian and Laplace distributions, which are important for understanding privacy-preserving mechanisms, and are used through the thesis.
- Section 2.3 discusses various distance measures between probability distributions, which are used for quantifying the differences and similarities in privacy contexts.

In Chapter 3 we discuss some classical approaches to privacy and anonymization (or their versions). Specifically:

- Section 3.1 and Section 3.2 introduce k-anonymity and Post Randomization (PRAM) providing mathematical formalizations and examples.
- Section 3.3 presents k-PRAM, a combination of the two methods discussed in Section 3.1 and Section 3.2, detailing its approach to improving data utility.

- Section 3.4 and Section 3.5 introduce noise addition and formally defines k-noise, a method of noise injection used to control group sizes and improving data utility with no bias.
- Section 3.6 provides experimental results, comparing the effectiveness of the discussed methods using a real-world dataset.

In Chapter 4 we introduce the fundamental concept of Differential Privacy, its definition, and its properties. To be more specific:

- Section 4.2 introduces the basic definition of differential privacy, explaining its fundamental principles and the role of the privacy budget (ε).
- Section 4.3 discusses the use of Laplace noise in differential privacy and proves that the Laplace mechanism satisfies the differential privacy definition.
- Section 4.4 extends the concept to approximate differential privacy, which allows for the use of normal distribution as noise.
- Section 4.5 highlights various properties of differential privacy, such as closure under post-processing and group privacy, and presents proofs and examples to illustrate these properties.

In Chapter 5 we examine differential privacy from the standpoint of data utility, exploring a range of mechanisms and applications. To be more specific:

- Section 5.2 discusses various sensitivity measures, including the global, local, and smooth sensitivity, and introduces the concept of general sensitivity. We show that the latter concept leads to improvement in data utility.
- Section 5.3 introduces the Mixed Noise Mechanism (MNM), detailing its formulation, theoretical proofs, and practical implementation.
- Section 5.4 presents two blocking algorithms designed to reduce the addition of noise while maintaining considerations of privacy and maximizing data utility.
- Section 5.5 introduces the bounded Laplace mechanism, ensuring realistic data entries in privacy-preserving queries.
- Section 5.6 compares pre-processing and post-processing approaches, analyzing their impacts on privacy and data utility for statistical estimators.
- Section 5.7 examines the challenges and solutions for maintaining accurate confidence intervals with privatized data.
- Section 5.8 discusses an alternative distance measure between probability distributions, called zero-concentrated differential privacy. Furthermore, it examines the implications of this measure for privacy and utility guarantees.

In Chapter 6 we explore how differentially privacy behaves in a time series setting. To be more specific:

- In Section 6.2, we formulate the privacy leakage problem in the time series setting. We distinguish between user-level and event-level privacy goals and consider different attack and data release scenarios. Additionally, we introduce the Vector Autoregressive (VAR) time series model.

- In Section 6.3, we present our theoretical results on privacy leakage in the time series context. The main results are Theorems 6.3.4 to 6.3.8, which provide formulas for privacy leakage under various attack and data release scenarios. These formulas depend on the sensitivity of the query and the model parameters. Despite their complexity, they can be easily calculated once the model parameters are estimated.

In Chapter 7 we explore differential privacy in the context of machine learning. Specifically, we determine the learning rate sequence that guarantees differential privacy in the Stochastic Gradient Descent algorithm. We provide the proofs for privacy bounds and for the rates of convergence. The main results are Theorem 7.6.1 (privacy bounds) and Theorem 7.7.1 (convergence of the algorithm).

We finish with some potential topics for future research in Chapter 8.

`R` codes can be found in Chapter A.

1.4 Thesis Contributions

This thesis presents several contributions to the field of differential privacy from both probabilistic and statistical theory perspective. The main contributions are as follows:

Mathematical framework

One of the most significant contributions of this thesis is the alignment of the language and methods used in computer science with those in probability and statistics. This process highlighted the notable challenge of bridging the terminology and methodology gaps between these two disciplines.

In our research, we identified numerous issues in the proofs found in the computer science literature. These included incorrect statements and incomplete arguments. To address these issues, we undertook a comprehensive effort to translate into a proper mathematical language and when necessary rewrite these proofs. We re-wrote many of the proofs that appeared in the computer science literature, making sure that they are correct from the mathematical point of view.

New techniques and results

Chapter 3. In Chapter 3 we introduce two novel contributions to the field of statistical disclosure control:

- k-PRAM: A method that combines the strengths of k-anonymity and absolute privacy measures, and Post Randomization (PRAM) to enhance data utility while ensuring privacy.

- k-noise: The k-noise method creates a novel bridge between statistical disclosure methods and noise injection techniques. This allows for comparative analysis of two distinct approaches to data anonymization and privacy preservation. Theorem 3.5.1 establishes a lower bound on k-noise and demonstrates equivalency to k-anonymity while significantly enhancing data utility. This theorem is an important result as it ensures that the k-noise method guarantees privacy comparable to k-anonymity, but with better data utility.

  The k-noise method effectively improves the utility of anonymized data. By adding controlled noise to grouped data, k-noise maintains group sizes in the transformed data set, thereby preserving data utility and ensuring lower bias in comparison to traditional methods, as well as k-PRAM. Through extensive experiments and simulations, the chapter demonstrates the practical effectiveness of k-noise. The results indicated that k-noise not only meets theoretical privacy guarantees but also performs better in real-world data scenarios.

  The results on k-PRAM and k-noise are published in [7].

Chapter 4. This chapter provides foundations on Differential Privacy and is based primarily on the existing literature. Some proofs were corrected and written in the proper mathematical language.

Chapter 5. This chapter includes many novel and original contributions to the field of differential privacy, from the data utility point of view. This line of research did not seem to be present in the literature (which focuses primarily on privacy guarantees).

- Section 5.2.2 contains a novel, practical approach to calculate sensitivity. It is based on the author's original work. Theorem 5.2.17 and its consequences is the most important result there.

- Section 5.2.1 is based on existing literature. However, some results were incorrect and they were stated and proven in the correct form.

- Section 5.3 introduces Mixed Noise Mechanism. It improves data utility. The most important result is Theorem 5.3.2. This section is based on the author's original work, [8].

- Section 5.4 includes a novel blocking algorithm, that improves data utility. See Theorem 5.4.3. It is based on the author's original work.
- In Section 5.6 we compare theoretically and numerically pre-processing and post-processing from the data utility point of view. It is based on the author's original work.
- Section 5.7 deals with differentially private confidence intervals with Theorem 5.7.2 as the most important result. The entire section is the author's original contribution.

Chapter 6. This chapter deals with the privacy in time series. The entire chapter is based on the author's original work with several new theorems.

Chapter 7. This chapter deals with the privacy in Stochastic Gradient Descent. The entire chapter is based on the author's original work with new theorems on privacy and convergence. These results are extendable to other optimization algorithms such as Coordinate Descent (used in LASSO procedure) or EM algorithm.

Some of the results of the thesis have been published already, e.g. [7] and [6]. Many of the results in this thesis are being submitted: Mixed Noise Mechanism of Section 5.3 ([8]); blocking algorithm of Section 5.4; confidence intervals of Section 5.7. The big and original contributions of Chapter 6 and Chapter 7 are also being submitted

Practical contributions

Some of the research in this thesis led to a US patent and to practical studies for the Office of the Privacy Commissioner of Canada. Many of the algorithms have been already implemented in the industry.

Chapter 2

Mathematical foundations

In this chapter, we introduce the basic terminology and mathematical framework that will be used throughout this thesis. These foundations are important to better understand more advanced concepts and methodologies discussed in the subsequent chapters.

2.1 Notation and basic terminology

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, and let $(\mathcal{U}, \mathcal{B}, \rho)$ be a metric space. The simplest possible case that we can consider is $\mathcal{U} = \mathbb{R}$, or $\mathcal{U} = \mathbb{R}^d$, $d > 1$ or $\mathcal{U} = [0, 1]$. Let $\boldsymbol{X} = (X_1, \ldots, X_n) \in \mathcal{U}^n$, be a random sample of size n from a distribution $\mathbb{P}$. That is, a database is a random element with values in $\mathcal{D} = \mathcal{U}^n$. In mathematical terms, $\boldsymbol{X} : \Omega \to \mathcal{D}$. We will denote realizations of $\boldsymbol{X} = (X_1, \ldots, X_n)$ by $\boldsymbol{x} = (x_1, \ldots, x_n)$, and we will call $\boldsymbol{X}$ (or its realizations) a **database**.

The **Hamming distance** $d : \mathcal{U}^n \times \mathcal{U}^n \to \{0, \ldots, n\}$ between two databases is the number of records on which they differ. We say that two databases $\boldsymbol{x}$ and $\boldsymbol{y}$ are **neighbours** if $d(\boldsymbol{x}, \boldsymbol{y}) = 1$, and we will denote neighbours as $\boldsymbol{x} \sim \boldsymbol{y}$. We denote a database where the i^{th} records is removed by $\boldsymbol{X}_{(-i)} = (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$. Note that $\boldsymbol{X}_{(-i)}$ is a random element with values in $\mathcal{U}^{n-1}$, but we can consider it as a random element of $\mathcal{U}^n$ by adding an empty record.

A **query** is a function f from $\mathcal{D}$ into $\mathbb{R}^d$. Some examples of queries include:

- Assume that $\mathcal{D} = \mathbb{R}^n$. The identity query $\mathrm{Id} : \mathbb{R}^n \to \mathbb{R}^n$ is $\mathrm{Id}(\boldsymbol{x}) = \boldsymbol{x}$, for $\boldsymbol{x} \in \mathbb{R}^n$.
- Assume that $\mathcal{D} = \mathbb{R}^n$. The mean query $\mathbb{R}^n \to \mathbb{R}$ is $f(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^n x_i$, for $\boldsymbol{x} \in \mathbb{R}^n$.
- Assume that $\mathcal{D} = \mathbb{R}^n$. The median query $\mathbb{R}^n \to \mathbb{R}$ is $f(\boldsymbol{x}) = \mathrm{median}(\boldsymbol{x})$, for $\boldsymbol{x} \in \mathbb{R}^n$.

An important element of differential privacy is the sensitivity of a function, and in some cases, the sensitivity of the database. In order to introduce the formal definitions for the sensitivity, we need to define the L_1 and L_2 norms.

For $\boldsymbol{u} = (u_1, \ldots, u_d)$, and $\boldsymbol{v} = (v_1, \ldots, v_d) \in \mathbb{R}^d$ we denote

$$\|\boldsymbol{u} - \boldsymbol{v}\|_1 = \sum_{i=1}^{d} |u_i - v_i|$$

and

$$\|\boldsymbol{u} - \boldsymbol{v}\|_2 = \sqrt{\sum_{i=1}^{d} (u_i - v_i)^2} \ .$$

Definition 2.1.1 (Local sensitivity)**.** *Let $j = 1, 2$. The **(local) sensitivity** of a function $f : \mathbb{R}^n \to \mathbb{R}^d$ is*

$$\Delta_j^{(\text{local})} f(\boldsymbol{x}) = \max_{\boldsymbol{y}, \boldsymbol{x} \sim \boldsymbol{y}} ||f(\boldsymbol{x}) - f(\boldsymbol{y})||_j.$$

Definition 2.1.2 (Global sensitivity)**.** *Let $j = 1, 2$. The **(global) sensitivity of** a function $f : \mathbb{R}^n \to \mathbb{R}^d$ is*

$$\Delta_j f = \sup_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}, \boldsymbol{x} \sim \boldsymbol{y}} \|f(\boldsymbol{x}) - f(\boldsymbol{y})\|_j \ . \tag{2.1}$$

The local and global sensitivities are related by

$$\Delta_j f = \max_{\boldsymbol{x} \in \mathcal{U}} \Delta_J^{(\text{local})} f(\boldsymbol{x}). \tag{2.2}$$

Intuitively, the local sensitivity measures the variability of a specific database $\boldsymbol{x}$, while the global sensitivity measures the variability of $\boldsymbol{x}$ in relation to all possible databases. In other words, to calculate the local sensitivity we consider neighbours of our database $\boldsymbol{x}$, while the global sensitivity considers the entire population of all databases.

If $d = 1$, then $\|\boldsymbol{u} - \boldsymbol{v}\|_1 = \|\boldsymbol{u} - \boldsymbol{v}\|_2$. In this case, $\Delta_1(f) = \Delta_2(f)$, so we will simply write Δf.

Example 2.1.3. Assume that the data, $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, come from a population with the range $R = [-\Lambda_1, \Lambda_2]$, where $0 \leq \Lambda_1, \Lambda_2 < \infty$. For the sample mean query, $f(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} x_i$, the local and global sensitivities are

$$\Delta^{(\text{local})} f(\boldsymbol{x}) = \max_{j=1,\ldots,n} \left| \frac{\sum_{i=1}^{n} x_i}{n} - \frac{\sum_{i=1, \text{i} \neq \text{j}}^{n} x_i}{n} \right| = \frac{1}{n} \max_{j=1,\ldots,n} |x_j| \ ,$$

$$\Delta f = \frac{\max\{\Lambda_1, \Lambda_2\}}{n} \ .$$

(Note that formally we should divide $\sum_{i=1,\mathrm{i}\neq\mathrm{j}}^{n} x_i$ by $n-1$, but this is a minor modification, especially if n is large). For the function $f(\boldsymbol{x}) = \text{median}(x_1, \ldots, x_n)$, assuming n is odd and $m = \frac{n+1}{2}$, the local and global sensitivities are:

$$\Delta^{(\text{local})} f(\boldsymbol{x}) = \max(x_{m+1} - x_m, x_m - x_{m-1}) \ ,$$

$$\Delta f = \Lambda_2 + \Lambda_1 \ .$$

Indeed, we can have a database of size, say, 11, with the first five entries equal to $-\Lambda_2$, and the remaining entries equal to Λ_1.

Finally, for identity function $\text{Id}(\boldsymbol{x}) = \boldsymbol{x}$, we use the Euclidean norm. Here, an issue arises since the original database is of dimension n, while the neighbour is of dimension $n-1$, and we cannot calculate $\boldsymbol{x} - \boldsymbol{y}$. To mitigate this issue, we assume that one record is removed from $\boldsymbol{x}$, replacing it (at random) with either $-\Lambda_1$ or Λ_2. Then,

$$\Delta^{(\text{local})} f(\boldsymbol{x}) = \max_{j=1,\ldots,n} \max\{|x_j - \Lambda_2|, |x_j + \Lambda_1|\} \ ,$$

$$\Delta f = \Lambda_1 + \Lambda_2 \ .$$

Example 2.1.4. Assume that the data, $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, come from a population with the range $R = [0, \Lambda]$, and mean μ. Define:

$$\bar{\boldsymbol{x}}_{i,j} = \frac{1}{n} \sum_{k=i+1}^{j} x_k \ , \quad \bar{\boldsymbol{x}} := \bar{\boldsymbol{x}}_{1,n} \ , \quad \bar{\boldsymbol{x}}_{-j} = \frac{1}{n} \sum_{\substack{k=1 \\ k\neq j}}^{n} x_k \ .$$

For the sample variance query, $f(\boldsymbol{x}) = \dfrac{\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})}{n-1}$, the local sensitivity is

$$\begin{aligned}
\Delta^{(\text{local})} f(\boldsymbol{x}) &= \max_{j=2,\ldots,n} \left\{ \frac{\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})}{n-1} - \frac{\sum_{i=1,i\neq j}^{n}(x_i - \bar{\boldsymbol{x}}_{-j})^2}{n-1} \right\} \\
&= \max_{j=1,\ldots,n} \left\{ \frac{x_j^2(1-1/n)}{n-1} - \frac{2x_j \bar{\boldsymbol{x}}_{-j}}{n-1} \right\} \\
&\overset{n \text{ large}}{\approx} \max_{j=1,\ldots,n} \left\{ \frac{x_j^2}{n} - \frac{2x_j\mu}{n} \right\} \ .
\end{aligned}$$

The last approximation stems from the law of large numbers. Furthermore, the global sensitivity is proportional to Λ^2/n.

Let $\mathcal{E}$ be another metric space. A **response mechanism** is a function $Q : \mathcal{D} \times \mathcal{E} \to \mathbb{R}^d$. Typically, the response mechanism is associated with a query f, and hence will be denoted by Q_f. Intuitively, Q perturbs the database $\boldsymbol{x} \in \mathcal{D}$ or a query with a "noise" $z \in \mathcal{E}$. In principle, Q can be an arbitrary function, but we introduce two types of functions that play an important role.

1. Let $\mathcal{E} = \mathcal{D}$ and assume that $\mathcal{D}$ is equipped with addition. Let $f : \mathcal{D} \to \mathbb{R}^d$ be a query. A **sanitized response (SR) mechanism** $S_f : \mathcal{D} \times \mathcal{D} \to \mathbb{R}^d$ is defined as

$$S_f(\boldsymbol{x}, \boldsymbol{z}) = f(\boldsymbol{x} + \boldsymbol{z}) \ , \quad \boldsymbol{x}, \boldsymbol{z} \in \mathcal{D} \ . \tag{2.3}$$

2. Let $\mathcal{E} = \mathbb{R}^d$ and let $f : \mathcal{D} \to \mathbb{R}^d$ be a query. An **output perturbation (OP) mechanism** $O_f : \mathcal{D} \times \mathbb{R}^d \to \mathbb{R}^d$ is defined as

$$O_f(\boldsymbol{x}, z) = f(\boldsymbol{x}) + z \ , \quad \boldsymbol{x} \in \mathcal{D}, z \in \mathbb{R}^d \ . \tag{2.4}$$

We note that for the identity query, both mechanisms agree. We include two drawings that illustrate the differences between the two mechanisms, and how they operate in Figure 2.1 and Figure 2.2.

Remark 2.1.5. We note that the Output Perturbation Mechanism is also referred to as Post-Processing, while the Sanitized Response Mechanism is also referred to as Pre-Processing.

Remark 2.1.6. In the case of the sanitized response, the "noise" $\boldsymbol{z}$ is of the same dimension as the database and will be denoted by $\boldsymbol{z}$. In the case of the output perturbation mechanism, the "noise" will typically (but not necessary) be one-dimensional and hence denoted by z. When speaking of noise addition in general terms, we will use the z notation. Furthermore, we will use a generic notation Q_f when speaking about sanitization or output perturbation in general terms.

Let Z be a random element with values in $\mathcal{E}$, that is, $Z : \Omega \to \mathcal{E}$. The type of noise Z (i.e., its distribution) will influence privacy. Set first $\mathcal{E} = \mathcal{D}$, and recall the convention from Remark 2.1.6. We will consider a **(randomized) sanitized response mechanism**, that is, a random element $S_f(\boldsymbol{x}, \boldsymbol{Z})$. Formally speaking, $\boldsymbol{Z} : \Omega \to \mathcal{D}$ and hence $S_f(\boldsymbol{x}, \boldsymbol{Z})$ is a map from $\mathcal{D} \times \Omega$ to $\mathbb{R}^d$. As such, in the case of general statements for arbitrary noise, we will write "a mechanism S_f", while when one needs to specify a specific noise, we will write: "a randomized mechanism $S_f(\cdot, \boldsymbol{Z})$". Likewise, $O_f(\cdot, Z)$ is a **(randomized) output perturbation mechanism**.

In other words, sanitization is nothing more than adding noise to each element of the database. A sanitized response mechanism corresponds to running a query on a database perturbed by noise. Such mechanisms correspond to **Privacy-Preserving Data Publishing (PPDP)** or **Pre-Processing**. On the other hand, the output perturbation mechanism corresponds to **Privacy-Preserving Data Mining (PPDM)** or **Post-Processing**.

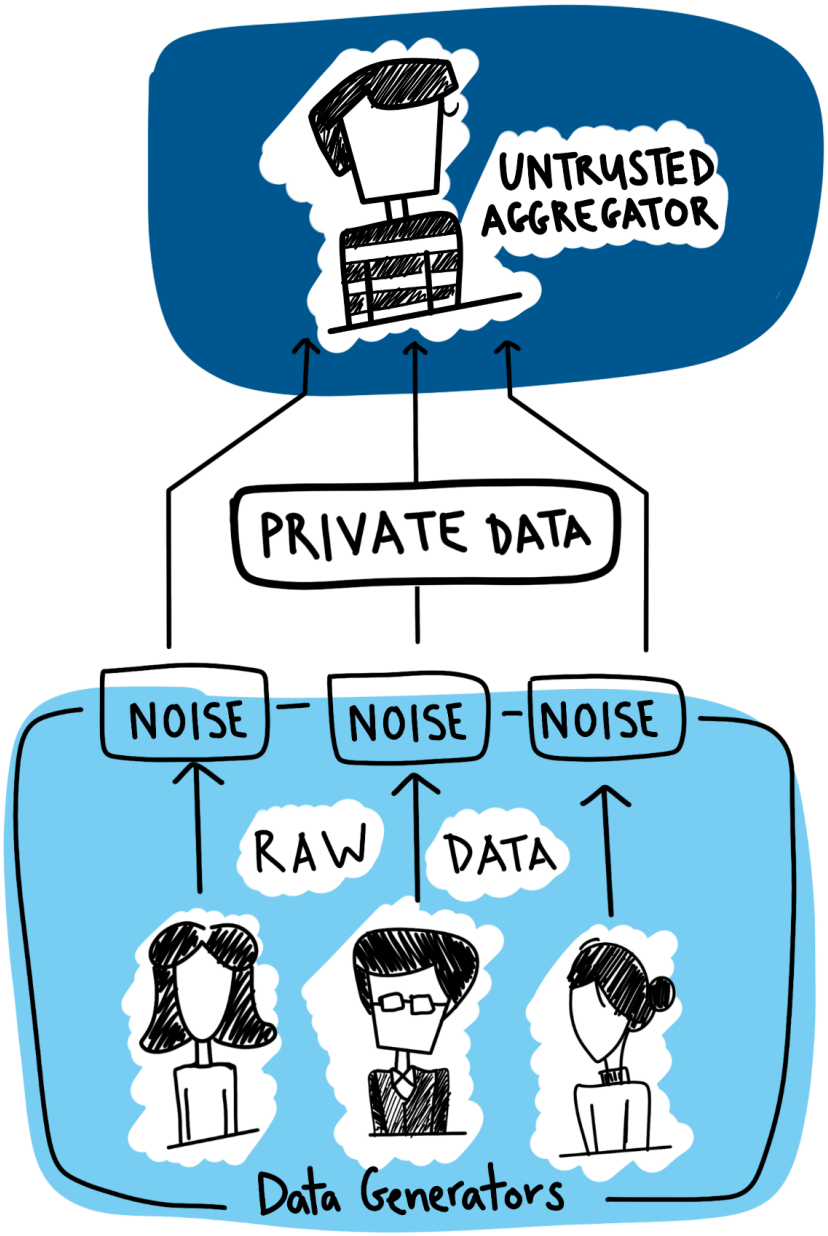

Figure 2.1: Drawing: sanitized response mechanism
In this drawing, illustrated by the author, the sanitized response mechanism is outlined. In this setting, noise is added to the individuals' data points before being shared to a untrusted aggregator. In this way, noise is added to the data *pre*-processing.

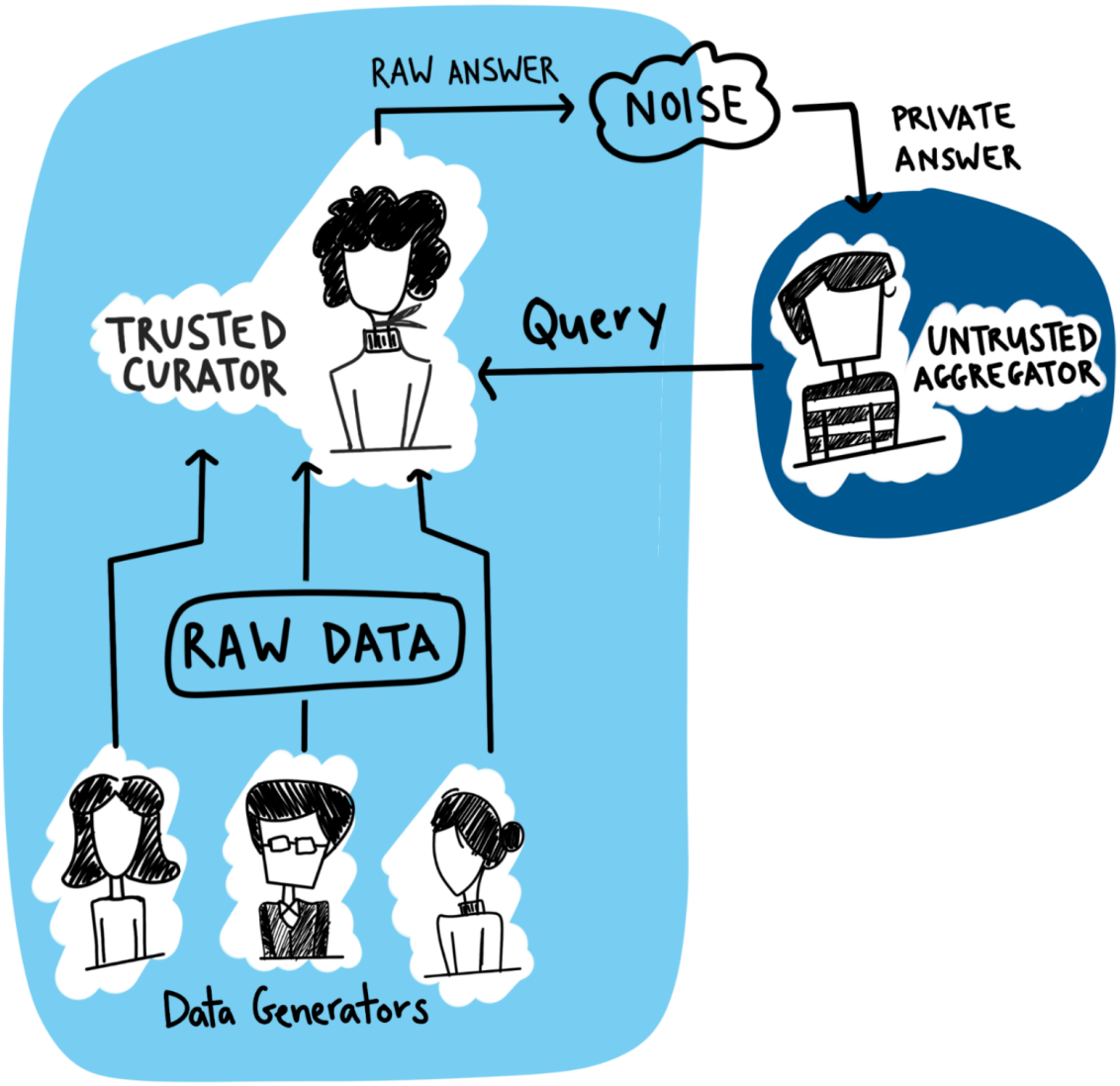

Figure 2.2: Drawing: output perturbation mechanism
In this drawing, illustrated by the author, the output perturbation mechanism is outlined. In this setting, a trusted curator receives a query from the untrusted aggregator. They collect the raw data, calculate the statistic, and then add noise to the query before returning the answer. In this way, noise is added *post*-processing of the data.

2.2 Probability distributions

Let Z be a random variable. By $g_Z(x)$, we will denote its density (when it exists) at point x. Alternatively, when convenient, we will use the notation $g(x; Z)$ to avoid a cumbersome subscript . If $(\boldsymbol{Z}_1, \boldsymbol{Z}_2)$ is a random vector, we will denote its density by $g_{(\boldsymbol{Z}_1,\boldsymbol{Z}_2)}(\boldsymbol{x}_1, \boldsymbol{x}_2)$. If there is no risk for confusion, we will drop the subscript. The density notation will be specified precisely in each chapter.

Two probability distributions will play a major role in our work. First, we will need a **normal** random variable denoted by $\mathcal{N}(\mu, \sigma^2)$. Second, we will need the **Laplace** (or double exponential) **distribution**, which is not widely used in statistics but will be crucial in this thesis.

Definition 2.2.1 (Normal Distribution)**.** *The Gaussian distribution is a continuous probability distribution for a real-valued random variable. The probability density function takes the form*

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{ -\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \right\},$$

where μ is the mean and σ is the standard deviation.

The central moments, for any non-negative integer p, are

$$\mathbb{E}[(X-\mu)^p] = \begin{cases} 0 & \text{if } p \text{ is odd ,} \\ \sigma^p (p-1)!! & \text{if } p \text{ is even .} \end{cases}$$

Definition 2.2.2 (Laplace Distribution on $\mathbb{R}$)**.** *The Laplace (double exponential) distribution is a continuous probability distribution for a real-valued random variable. The probability density function takes the form*

$$g(x) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b} \right),$$

where μ is the mean and $b > 0$ is the scale parameter. We will write Laplace(μ, b) *to denote a random variable with such distribution. If the mean is zero, we will write* Laplace(b)*.*

The variance is given by $\sigma^2 = 2b^2$. The moments about the mean, μ_n, are given by

$$\mu_n = \begin{cases} n! b^n & \text{for } n \text{ even ,} \\ 0 & \text{for } n \text{ odd .} \end{cases}$$

The characteristic function can also be used to compute the moments:

$$\phi(t) = \frac{1}{2b} \int_{-\infty}^{\infty} e^{itx} e^{-|x-\mu|b} dx = \frac{e^{i\mu t}}{1 + b^2 t^2} .$$

Finally, we also have the following scaling property.

Lemma 2.2.3. *If $Z' \sim \text{Laplace}(1)$, then $bZ' \sim \text{Laplace}(|b|)$.*

We will refer to Z' as a standard Laplace random variable.

Definition 2.2.4 (Laplace Distribution on $\mathbb{R}^p$)**.** *The multivariate Laplace distribution is a continuous probability distribution on $\mathbb{R}^p$. The probability density function takes the form*

$$g(\boldsymbol{x}) = \frac{1}{(2b)^p} \exp\left(- \frac{\|\boldsymbol{x} - \boldsymbol{\mu}\|_1}{b} \right),$$

where $\boldsymbol{\mu} \in \mathbb{R}^p$ is the mean vector, $b > 0$ is the scale parameter, and $\| \cdot \|_1$ is the L_1-norm on $\mathbb{R}^p$.

If $\boldsymbol{Z}$ is a random vector with a Laplace distribution, we call it standard if $\boldsymbol{\mu} = 0$ and $b = 1$. We note that

$$\mathbb{E}[\|\boldsymbol{Z}\|_1^2] = p^2 + p \ , \quad \mathbb{E}[\|\boldsymbol{Z}\|_2^2] = 2p =: c_Z \ . \tag{2.5}$$

2.3 Distance between probability distributions

Let P and Q be two probability measures on $(\Omega, \mathcal{F})$. We present different ways to measure the distance between them.

Definition 2.3.1. *The total variation distance is defined as*

$$d_{\text{TV}}(P, Q) := \sup_{A \in \mathcal{F}} |P(A) - Q(A)| \ .$$

Definition 2.3.2. *The relative entropy (Kullback-Leibler) distance is defined as*

$$D_{\text{KL}}(P \| Q) := \mathbb{E}_P \left[\log \frac{\mathrm{d}P}{\mathrm{d}Q} \right] \ .$$

Definition 2.3.3. *The max-divergence distance is defined as*

$$D_\infty(P \| Q) := \sup_{A \in \mathcal{F}} \log \frac{P(A)}{Q(A)} \ .$$

Definition 2.3.4. *Let $\delta > 0$. The approximate max-divergence distance is defined as*

$$D_\infty^\delta(P \| Q) := \sup_{A \in \mathcal{F} : P(A) > \delta} \log \frac{P(A) - \delta}{Q(A)} \ .$$

Definition 2.3.5. *Let $\alpha > 0$. The α-Rényi divergence is defined as*

$$D_\alpha(P \| Q) = \frac{1}{\alpha - 1} \log \mathbb{E}_P \left[\left(\frac{dP}{dQ} \right)^{\alpha - 1} \right] \ .$$

Definition 2.3.6. *Let $\gamma > 1$. The E_γ-divergence distance is defined as*

$$E_\gamma(P\|Q) := \sup_{A\in\mathcal{F}}\{P(A) - \gamma Q(A)\} .$$

There are some classical inequalities between these distances:

$$d_{\mathrm{TV}}(P,Q) \le \sqrt{D_{\mathrm{KL}}(P\|Q)/2} , \tag{2.6a}$$
$$D_{\mathrm{KL}}(P\|Q) \le D_\alpha(P\|Q) \le D_\infty(P\|Q) , \tag{2.6b}$$
$$1 - \gamma(1 - d_{\mathrm{TV}}(P,Q)) \le E_\gamma(P\|Q) . \tag{2.6c}$$

Example 2.3.7. Assume that P and Q are the normal laws with means μ_P, μ_Q, respectively, and a common variance $\sigma^2 = \sigma_P^2 = \sigma_Q^2$. The distances are calculated as follows:

- For the total variation distance, we have

$$d_{\mathrm{TV}}(P,Q) = 4\Phi\left(\frac{\mu_P - \mu_Q}{2\sigma^2}\right) - 2 ,$$

  where Φ is the standard normal cumulative distribution function.

- For the KL-distance, we have

$$D_{\mathrm{KL}}(P\|Q) = \frac{(\mu_P - \mu_Q)^2}{2\sigma^2} .$$

- For the D_α distance, we have

$$D_\alpha(P\|Q) = \alpha\frac{(\mu_P - \mu_Q)^2}{2\sigma^2} .$$

  Note that $\lim_{\alpha\to 1} D_\alpha = D_{\mathrm{KL}}$, and additionally, $\lim_{\alpha\to\infty} D_\alpha = D_\infty$. Hence,

$$D_\infty(P\|Q) = \infty .$$

Chapter 3

Techniques for disclosure control

In this chapter, we review some of the existing techniques used in disclosure control. All these techniques involve perturbation of the original database $\boldsymbol{X} = (X_1, \ldots, X_n)$. The transformation can be achieved through various methods, such as:

- Grouping/generalization (k-anonymity, see Section 3.1);
- Post Randomization (PRAM, see Section 3.2);
- Combination of grouping and Post Randomization (k-PRAM, see Section 3.3);
- Noise addition (see Section 3.4; Differential Privacy is discussed in Chapter 4);
- Combination of grouping and noise addition (k-noise, see Section 3.5).

Each technique will be introduced with a brief explanation, followed by detailed descriptions and some examples to illustrate their application. We cite the proper references in each section separately. The contents of Sections 3.3 and 3.5 comes from [7] and it is the original work of the authors of the thesis.

3.1 k-anonymity

The method of k-anonymity was formally introduced by Latanya Sweeney in [44], although the concept was first mentioned in 1986 by Tore Delanius. A release of a database is considered to be k-anonymous if the information for each person contained in the database cannot be distinguished from at least $k-1$ other individuals, whose information is also contained in the released database.

We formalize the concept mathematically and provide an example.

Definition 3.1.1. *Consider the database* $\boldsymbol{X} = (X_1, \ldots, X_n)$*. Fix an integer* k*. The dataset is divided into* m *subgroups. These subgroups are called equivalence classes. Each individual belongs to one and only one equivalence class. An anonymized dataset* $\boldsymbol{Y}$ *provides* k*-anonymity if, for each individual* Y_j *in the given equivalence class, there exists at least* $k-1$ *other individuals in the same class with identical values.*

In other words, the probability that a particular individual is identified is at most $1/k$. This probability does not depend on the original database, thus k-anonymity is viewed as an *absolute* measure of privacy.

The larger the value of k, the higher the level of privacy achieved. However, to achieve a large k, one needs either a large population or a high level of generalization and suppression. These data transformations can negatively impact data utility.

Example 3.1.2. We illustrate two examples of k-anonymity using the same database. The database used for this example is fictitious.

Customer Age	Transaction Amount($)
32	75.50
45	120.00
27	16.75
19	85.00
31	50.00
25	40.50

Table 3.1: k-anonymity example
Company A customer transactions.

Based on this database, if we know that John Smith is 31 years old, we immediately know his Transaction Amount.

To illustrate k-anonymity with respect to one variable, the Company A can group the Customer Age into the intervals $[18-30]$ and $[31-45]$. They would then release the following database to ensure k-anonymity, with $k = 3$, for the variable Customer Age.

Here, even if we know that John Smith is 31 years old, our chance to guess his Transaction Amount is 1/3.

To illustrate k-anonymity with respect to both variables, Company A can group the Transaction Amount into the intervals $[0-50]$, and $[51-100]$. They would then release the following database to ensure k-anonymity, with $k = 3$, for both variables in the dataset.

Customer Age	Transaction Amount($)
(31-45)	75.50
(31-45)	120.00
(18-30)	16.75
(18-30)	85.00
(31-45)	50.00
(18-30)	40.50

Table 3.2: 3-anonymity for 1 variable
3-anonymization with respect to the variable Age.

Customer Age	Transaction Amount($)
(31-45)	(51-100)
(31-45)	(51-100)
(18-30)	(0-50)
(18-30)	(51-100)
(31-45)	(0-50)
(18-30)	(0-50)

Table 3.3: 3-anonymity for 2 variables
3-anonymization for the variables Age and Amount.

3.2 PRAM

Post Randomization (PRAM) was formally introduced in [24]. The method applies to categorical data, that is, when the possible realizations of the random variables X_j lie in the set $\{a_i, i = 1, \ldots, M\}$, where a_i are real values. The basic idea is as follows: each of X_j's is transformed into Y_j according to the given transition probabilities:

$$p_{kl} = \mathbb{P}(Y_j = a_l \mid X_j = a_k) \ .$$

The disclosure risk in PRAM is measured through *posterior odds*, that is, the relative probability that a rare score in the perturbed dataset $\boldsymbol{Y}$ corresponds with a rare score in the original dataset $\boldsymbol{X}$. These posterior odds should be small. Data utility is measured by the increase in the variance of the estimates due to the measurement error introduced by PRAM. Theoretical formulas for the variances are provided.

Example 3.2.1. We can illustrate this concept with a simple example. Suppose that the variable X_j represents gender of jth person with the possible values of 0 if you are male and 1 if you are female. PRAM can be applied to the gender variable so that $p_{kk} = 0.8$. Assume the database contains 1000 people, consisting of 500 men and 500 women. The expected perturbed database will also contain 500 men and women, but 100 men and 100 women would have had their gender swapped.

3.3 k-PRAM

This method can be viewed as a combination of k-anonymity and PRAM. It was introduced in [7], with the goal of maximizing data utility in statistical disclosure methods.

Similar to k-anonymity, we want to divide the dataset $\boldsymbol{X}$ into m subgroups in such a way that each subgroup (equivalence class) has at least k entries. If the original dataset is inhomogeneous, with large variability and outliers, this may not be possible to achieve. However, if the original data follow a specific probability distribution, the subgroups can be selected so that the expected number of entries in each of them is at least k. To be more specific, assume that X_j's are real-valued. Let $X_{(1)} \leq \ldots \leq X_{(n)}$ be the order statistics of $\boldsymbol{X}$ and define $\text{Range}(\boldsymbol{X}) = X_{(n)} - X_{(1)}$. Let $G_1, \ldots, G_m$ be m consecutive intervals (subgroups) of equal length

$$|G| := |G_i| = \frac{\text{Range}(\boldsymbol{X})}{m} \ , \quad i = 1, \ldots, m \ .$$

That is, $G_1 = [X_{(1)}, X_{(1)} + |G|)$, $G_2 = [X_{(1)} + |G|, X_{(1)} + 2|G|)$, and so on. We require that the expected size of each subgroup is at least k:

$$\mathbb{E}\left[\sum_{j=1}^{n} \mathbb{1}\{X_j \in G_i\}\right] \geq k \ .$$

After the data are grouped into the intervals $G_1, \ldots, G_m$, we apply randomization using PRAM to each of the individual subgroups G_i separately in such a way that the size of each subgroup remains constant, and hence the disclosure risk is at most $1/k$, as with k-anonymity.

Randomization can be applied in this way as a means of misleading would-be attackers or simply to maintain data formats. As we will see in Section 3.5, this is not desirable from the data utility point of view.

Example 3.3.1. Continuing with Example 3.2.1, we can extend it to k-PRAM. Suppose our database also contains Age, where the range of ages is $[20, 80]$. We want to split Age into subgroups so that at least k people exist in each subgroup. We then apply PRAM to each subgroup such that for any individual in a subgroup, their probability transition matrix is with respect to the subgroup and not the full range of Ages. So if the groups are split into $G_1 = [20, 40), G_2 = [40, 60), G_3 = [60, 80]$, each G_i has its own $p_{kl}(G_i)$. In this way, the data is perturbed but we can control the group size through k-anonymity.

3.4 Noise addition

PRAM and k-anonymity were conceived to limit disclosure risk from microdata. Noise addition can be viewed as a method to protect privacy, especially for continuous data. In particular, in the context of Differential Privacy (to be discussed thoroughly in Chapter 4), the goal is to limit disclosure risk from statistical queries.

The basic set-up is as follows. Let $\boldsymbol{X}$ be the database. Then a randomized dataset is defined by

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{Z}\,,$$

where $\boldsymbol{Z}$ is a vector of independent random variables that follow a particular distribution. In the language of Section 2.1, we consider the randomized sanitized response mechanism with the identity query f.

As indicated above, noise addition became a methodology in the context of Differential Privacy. In principle, there is no link between noise addition and classical k-anonymity. One of the links is provided by the k-noise methodology, to be introduced in the following section.

3.5 k-noise

This method can be viewed as a combination of k-anonymity and noise addition. The method was introduced in [7], as an extension of the concept formalized in Section 3.3. If the data are grouped, as with k-anonymity, and arbitrary noise is added to individual data points, there is no guarantee that the group sizes in the transformed dataset are preserved. However, with carefully prescribed noise addition, the group sizes in the transformed dataset can be controlled. As such, the disclosure risk can be similarly controlled as is the case with k-anonymity.

Once the privacy level is fixed, we can focus efforts on improving data utility. As opposed to the randomization within fixed intervals or groups, as described in Section 3.3, this novel approach does not introduce bias and hence has better data utility.

We divide the dataset into m groups G_i of the same length $|G| = |G_i| = 2\delta$ with some $\delta > 0$. This implies that in a 2δ-neighbourhood of any record $x \in \boldsymbol{X}$, we have at least k individuals:

$$\#\{j : |X_j - x| < 2\delta\} \geq k\,.$$

We note, however, that we cannot control the number of individuals in a δ-neighbourhood, $\#\{j : |X_j - x| < \delta\}$, as shown in Figure 3.1 below.

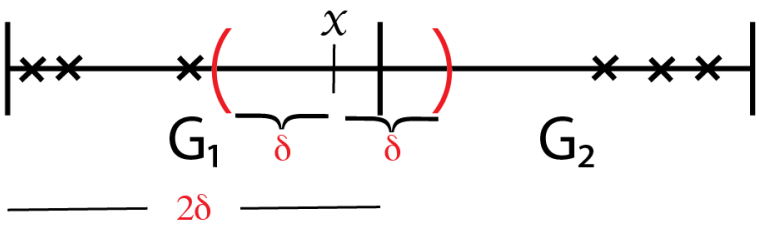

Figure 3.1: k-noise group
Graphical representation of the δ neighbourhood of a record x in the dataset.

Let $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ be a randomized dataset defined by

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{Z}\,,$$

where $\boldsymbol{Z} = (Z_1, \ldots, Z_n)$ is a vector of independent identically distributed random variables.

Uniformly distributed noise

We will assume in this section that Z_j, $j = 1, \ldots, n$, have a uniform distribution with a parameter $a > 0$. If for a particular group G_i, the data X_j are concentrated around its centre, then the choice $a = \delta$ guarantees, with a high probability, that

$$\#\{j : |Y_j - x| < 2\delta\} \geq k\,.$$

However, if this is not the case, the bound cannot be guaranteed. Thus, the theoretical bound must consider the worst-case scenario and be more conservative. The most conservative bound guarantees that by applying a uniform noise with parameter δ there exists at least $\frac{1}{2}k$ other individuals within a 2δ neighbourhood. Furthermore, we show that the underlying distribution of the dataset is not needed in order to guarantee this bound.

Theorem 3.5.1. *Let $\boldsymbol{X} = (X_1, \ldots, X_n)$ and $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{Z}$, where $\boldsymbol{Z} = (Z_1, \ldots, Z_n)$ is a vector of independent uniform random variables on $[-a, a]$ for $a > 0$. Let $\delta > 0$ and assume that for each $x \in [X_{(1)}, X_{(n)}]$ we have*

$$\#\{j : |X_j - x| < 2\delta\} \geq k\,.$$

Take $a = \delta$. Then

$$\mathbb{E}\left[\#\{j : |Y_j - x| \leq 2\delta\} \mid \boldsymbol{X}\right] > \frac{1}{2}\#\{j : |X_j - x| < 2\delta\} = \frac{1}{2}k\,.$$

Remark 3.5.2. We note that the expectation is calculated conditionally on the database $\boldsymbol{X}$, hence the database entries are treated as deterministic and the randomness is due to the noise $\boldsymbol{Z}$. Using the tower property of the conditional distribution we also obtain

$$\mathbb{E}\left[\#\{j : |Y_j - x| \leq 2\delta\}\right] > \frac{1}{2}\#\{j : |X_j - x| < 2\delta\} = \frac{1}{2}k\,.$$

Proof of Theorem 3.5.1. Let $A_j = -2\delta - X_j + x, B_j = 2\delta - X_j + x$. Then, using the properties of the uniform distribution,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{j=1}^{n} \mathbb{1}\{-2\delta < Y_j < 2\delta\} \mid \boldsymbol{X}\right] \\
&= \sum_{j=1}^{n} \mathbb{E}\left[\mathbb{1}\{-2\delta - X_j + x < Z_j < 2\delta - X_j + x\} \mid \boldsymbol{X}\right] \\
&= \frac{2\delta}{a}\sum_{j=1}^{n} \mathbb{1}\{-a < A_j, B_j < a\} + \sum_{j=1}^{n} \mathbb{1}\{A_j < -a, a < B_j\} \\
&\quad + \frac{1}{2a}\sum_{j=1}^{n}(a - A_j)\mathbb{1}\{-a < A_j, a < B_j\} + \frac{1}{2a}\sum_{j=1}^{n}(B_j + a)\mathbb{1}\{A_j < -a, B_j < a\}.
\end{aligned}$$

For $a = \delta$ the expressions above become

$$\begin{aligned}
&\sum_{j=1}^{n} \mathbb{1}\{x - \delta < X_j < x + \delta\} \\
&+ \sum_{j=1}^{n} \frac{(3\delta - x + X_j)}{2\delta}\mathbb{1}\{x - 3\delta < X_j < x - \delta\} \\
&+ \sum_{j=1}^{n} \frac{(3\delta + x - X_j)}{2\delta}\mathbb{1}\{x + \delta < X_j < x + 3\delta\}.
\end{aligned}$$

We split the last two terms as $J_1 + J_2 + J_3 + J_4$ with

$$J_1 := \sum_{j=1}^{n} \frac{(3\delta - x + X_j)}{2\delta} \mathbb{1}\{x - 2\delta < X_j < x - \delta\},$$
$$J_2 := \sum_{j=1}^{n} \frac{(3\delta - x + X_j)}{2\delta} \mathbb{1}\{x - 3\delta < X_j < x - 2\delta\},$$
$$J_3 := \sum_{j=1}^{n} \frac{(3\delta + x - X_j)}{2\delta} \mathbb{1}\{x + \delta < X_j < x + 2\delta\},$$
$$J_4 := \sum_{j=1}^{n} \frac{(3\delta + x - X_j)}{2\delta} \mathbb{1}\{x + 2\delta < X_j < x + 3\delta\} =: I_1 + I_2 + I_3.$$

Note that

$$J_1 + J_3 \geq \frac{1}{2} \sum_{j=1}^{n} \mathbb{1}\{x - 2\delta < X_j < x - \delta\} + \frac{1}{2} \sum_{j=1}^{n} \mathbb{1}\{x + \delta < X_j < x + 2\delta\} \ .$$

Ignoring J_2 and J_4, the expectation is bounded below by

$$\frac{1}{2} I_1 + J_1 + J_3 \geq \frac{1}{2} \#\{j : |X_j - x| < 2\delta\} \geq \frac{1}{2} k \ .$$

□

3.6 Experimental Results

In the first experiment, we illustrate that the although the bound obtained in Theorem 3.5.1 can be conservative, in reality it is close to the target value of k, and in many cases can exceed k. This is shown in Section 3.8 and Figure 3.3. We show experimental results using a public dataset consisting of 659 records with several categorical and numerical variables. We focus on one numerical variable of interest, Age, and aim to study the effects of data utility when comparing two methods of anonymization. In Section 3.8, the histogram of the original ages (since PRAM preserves the counts of the histogram) and Figure 3.3 shows the histogram for the noisy data where Z_j has the uniform distribution, Unif$[-\delta, \delta]$.

Using the same binning between histograms, we can see that the empirical distributions for both the original and the noisy datasets are nearly identical, and hence the data utility (measured by an arbitrary metric) is comparable. The difference between the k-PRAM and k-noise methods are illustrated on Figure 3.4.

With k-noise, the resulting distribution of ages is smoother, which suggests better utility and has the added benefit of further misleading would-be attackers. The light blue clusters show the inherent bias in the dataset when implementing k-PRAM, versus the smooth dark blue trend formed when implementing k-noise. Furthermore, we divide the Age variable into 12 groups, each spanning an interval of 5 years on the interval $[24, 79]$, and we can see from Table 3.4 that k-noise reduces the bias and error compared to k-PRAM.

Method	**Bias**	**MSE**	**RMSE**
k-PRAM	0.06881953	4.398563	2.097275
k-noise	0.03408935	2.060666	1.435502

Table 3.4: Utility results for k-PRAM & k-noise
Different utility measures to compare k-PRAM and k-noise methods.

To test this further, we employ the use of Monte Carlo simulations to get the expected number of ages, representing individuals, in a neighbourhood of an anonymized entry when applying k-noise. k-noise can be thought of as a "local" measure of k-anonymity, since the group is being compared to the neighbourhood of adjacent points. If this number of ages exceeds or equals the group size of the original entry, then we can determine they are adequately protected within a group. We are treating the underlying dataset as the baseline for comparison to k-noise. Our results far exceed our theoretical bound of $\frac{1}{2}k$ and demonstrates the effectiveness of this approach in practice.

3.7 Conclusion

By adjusting the noise level to achieve an expected minimum threshold k, we can improve the distribution of an anonymized variable over the more common approach of randomizing within fixed intervals to satisfy k-anonymity. This noise addition approach allows us to leverage the well-established concept of k-anonymity, which is easily understood and has well-established precedents for the threshold k. We believe this will enable us to fine-tune noise levels based on other statistical properties and make inroads towards bridging k-anonymity with differential privacy which will be introduced and studied extensively in the remainder of this thesis.

3.8 Figures

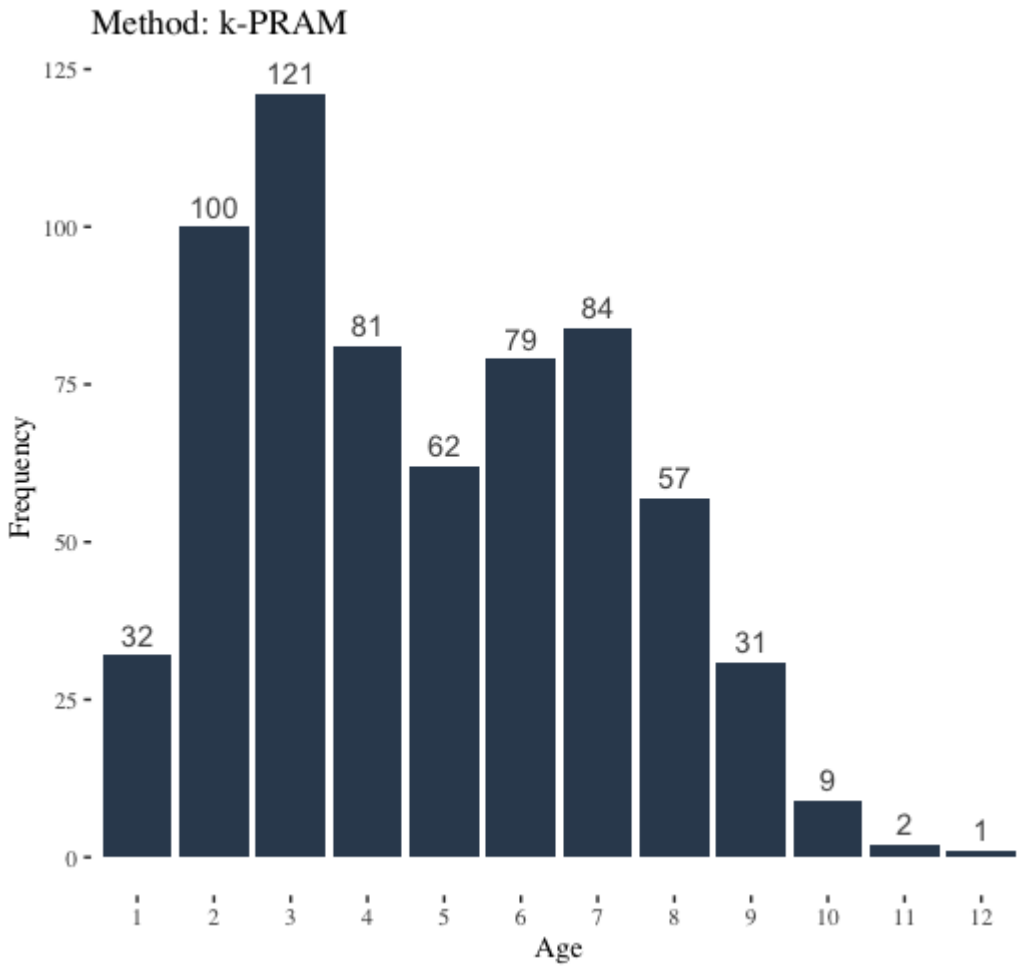

Figure 3.2: k-PRAM
Empirical distribution of randomized dataset via k-PRAM

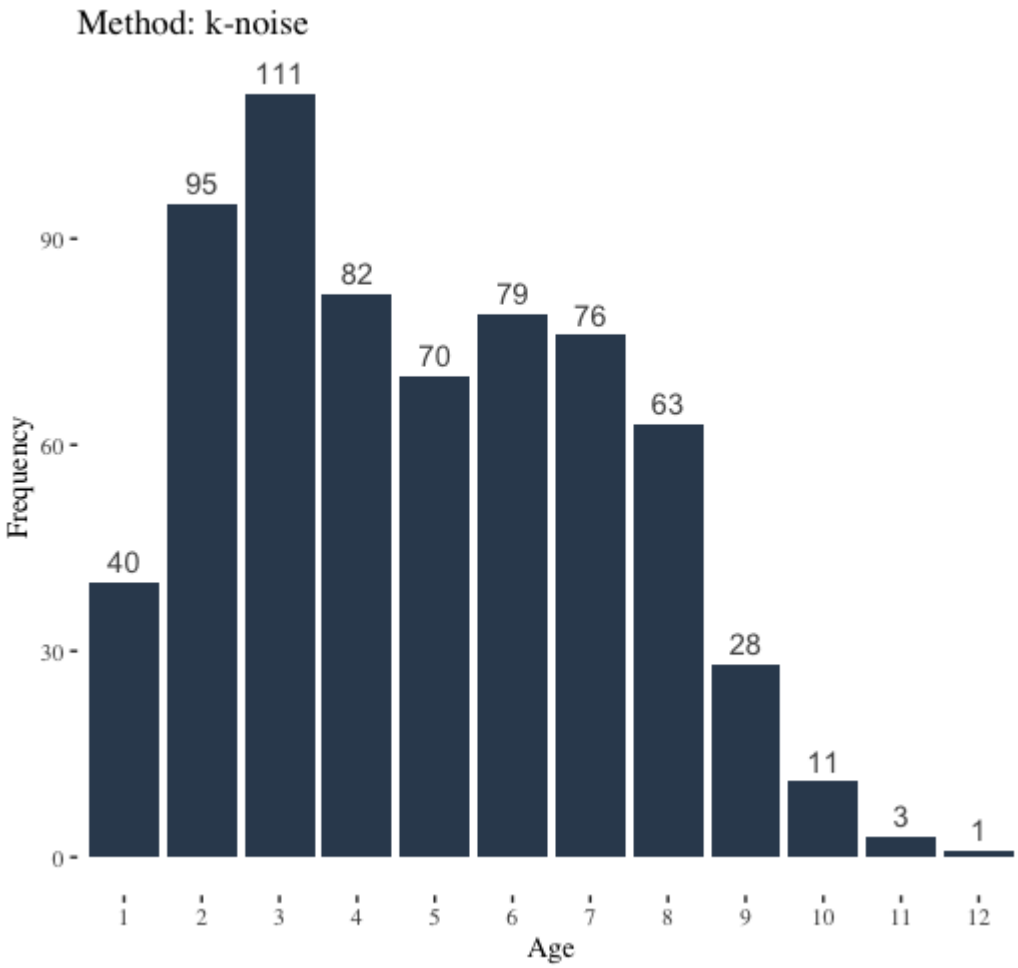

Figure 3.3: k-noise
Empirical distribution of randomized dataset via k-noise

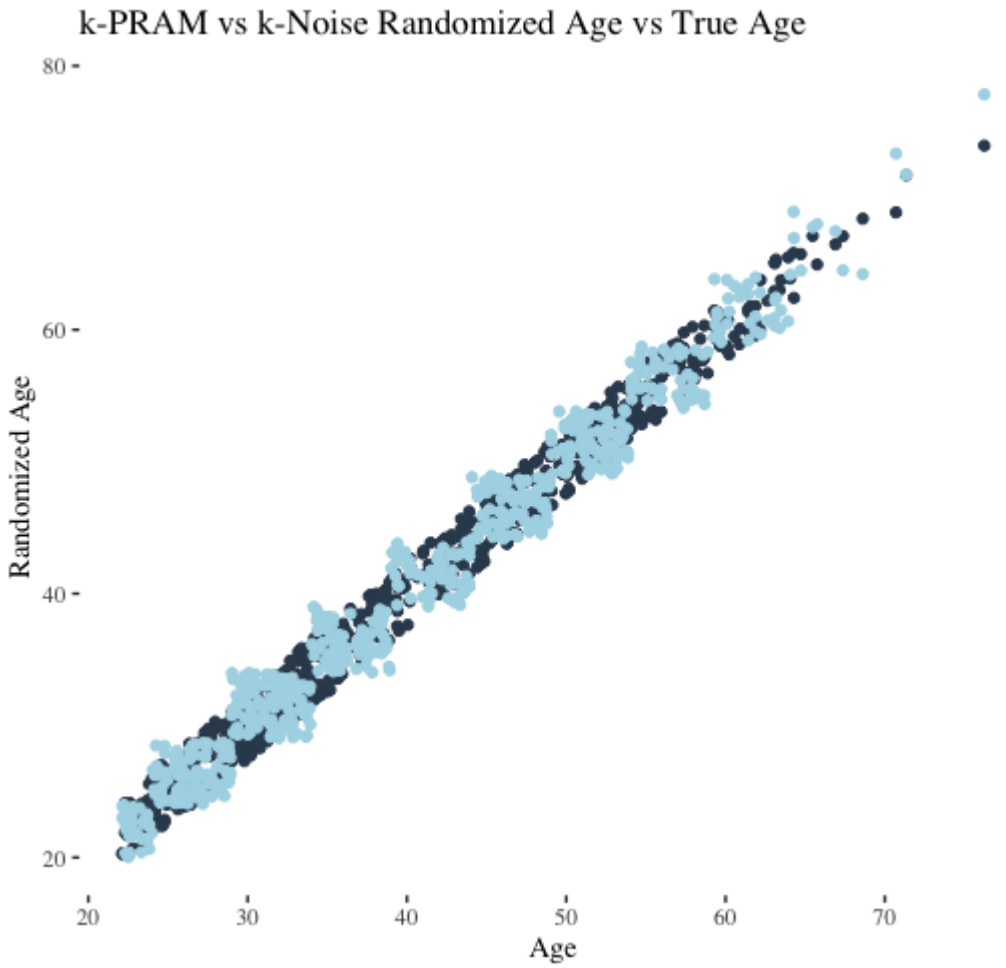

Figure 3.4: k-PRAM vs k-noise
Scatterplot of anonymized ages implementing k-PRAM and k-noise on the same database.

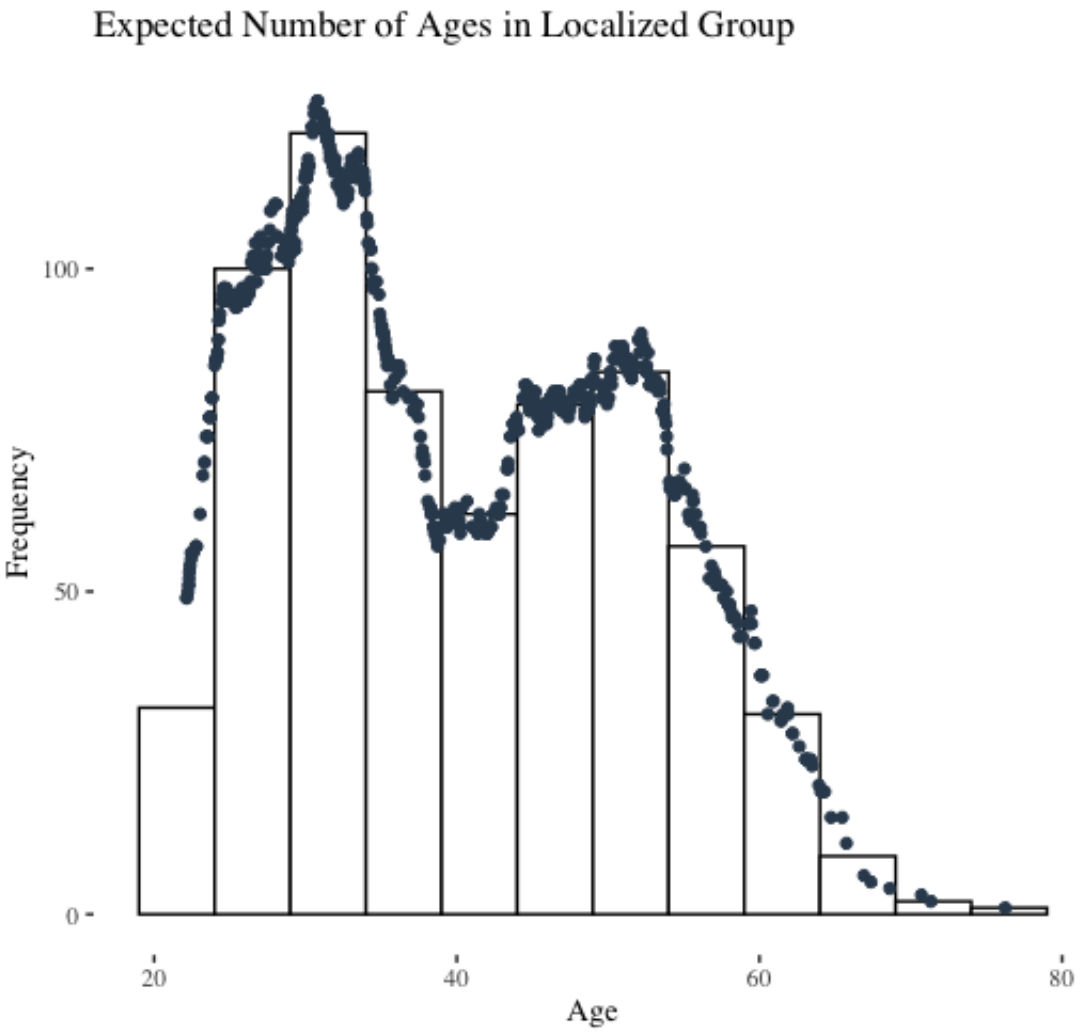

Figure 3.5: Expected group size
Expected number of records within a δ neighbourhood of [-2.5,2.5] years for each randomized record in $\boldsymbol{X}$.

Chapter 4

Differential privacy

Differential Privacy is a probabilistic guarantee that the inclusion of an individual in the database does not alter the outcome of a query on the database by more than a specific bound.

We start with the basic definition in Section 4.2. Next, it is shown (see Theorem 4.3.1) that a particular output perturbation mechanism with Laplace noise fulfills differential privacy. It turns out that other classical distributions, such as the normal distribution, violate the classical definition. As such, in recent years, differential privacy has became almost synonymous with adding noise following a Laplace distribution.

In Section 4.4 the concept of approximate differential privacy is introduced. This approximation allows for the inclusion of the normal distribution as noise. Section 4.5 highlights different closure properties of differential privacy.

The bulk of this chapter is based on the foundation work by [17], with several examples and proofs added by the author of the thesis.

4.1 Introduction

Differential privacy is a probabilistic guarantee that the inclusion of an individual in a database does not alter the outcome of a query on the database by more than a user chosen specified bound. It represents a robust framework for quantifying and managing the privacy of individuals in databases that undergo analysis. Formally introduced by Cynthia Dwork in 2005, it has since become a popular implementation choice in the field of data privacy. It provides mathematical guarantees against identity inference and data re-identification attacks.

The foundation of differential privacy is built upon several key concepts, which we introduce at a high level in this introduction. The **privacy budget**, known as **epsilon**

(ε), is an important parameter in a differentially private mechanism. It quantifies the allowable loss of privacy (or increase in privacy risk) when a query is answered using a database containing private or sensitive information. **A smaller value of epsilon offers stronger privacy guarantees**, limiting the information that can be inferred about any individual. However, **with stronger privacy guarantees comes reduced accuracy in the query results, namely a decrease in data utility**. The data utility issues will be discussed in great detail in Chapter 5. The **sensitivity function** refers to the maximum change a single individual's data can affect the output of a statistical query. In the context of differential privacy, it is an important parameter in the noise mechanism, measuring how much the result of a query function can vary when a single record in the database is altered. Since it is a parameter in the noise mechanism, it is essential for calibrating the noise added to the query response, contributing to the balance of data privacy vs. data utility.

Finally, there are two implementations of differential privacy: **local differential privacy** and **global differential privacy**. The former ensures privacy guarantees at the individual data point level. Each individual's data is randomized before it is collected by the aggregator. This corresponds to the sanitized response (SR) mechanism defined in (2.3). This approach is considered highly robust to privacy risks since the raw data is never used for statistical analysis; however, the high level of noise required often leads to significant reductions in data utility. Global differential privacy ensures privacy at the query level by first aggregating the data and then applying noise to the aggregated output before any output is shared. This corresponds to the output perturbation (OP) mechanism introduced in (2.4). This approach requires a trusted curator to hold the raw data, but generally provides better data utility.

We note in passing that global and local differential privacy should not be conflated with global and local sensitivity (See Definition 2.1.2). The global and local differential privacy is the implementation of the randomized mechanism, while global and local sensitivity refers to the way in which we measure the contribution of the individual in the database. For example, we can use the global differential privacy with global sensitivity (in this case, differential privacy guarantees will hold), or we can use global differential privacy with local sensitivity (here, differential privacy guarantees may be violated). We will discuss such issues in the next chapter.

4.2 Basic definition

We will refer to the notation introduced in Section 2.1. Recall that $f : \mathcal{D} \to \mathbb{R}^d$. Let $\mathcal{B}(\mathbb{R}^d)$ be the class of Borel sets on $\mathbb{R}^d$. Recall that we use the notation $Q = Q_f$ to denote either the sanitized response mechanism S_f or output perturbation mechanism O_f.

The general idea of differential privacy can be summarized as follows. Assume that

we have realizations of two neighbouring databases $\boldsymbol{x}$ and $\boldsymbol{y}$ and consider a randomized response mechanism $Q_f(\cdot, Z)$ acting on either $\boldsymbol{x}$ or $\boldsymbol{y}$. The mechanism is differentially private if the distance between the appropriate probability distributions of $Q_f(\cdot, Z)$ is bounded.

Definition 4.2.1 (Differential Privacy, DP)**.** *Let $\boldsymbol{X}$ be a database, a random element of $\mathcal{D}$. Let Z be a random element with values in a metric space $\mathcal{E}$. Let $\varepsilon > 0$. A randomized mechanism $Q : \mathcal{D} \times \mathcal{E} \to \mathbb{R}^d$ is ε-differentially private if, $\forall\ \boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$, satisfying $d(\boldsymbol{x}, \boldsymbol{y}) = 1$, we have*

$$\sup_{B \in \mathcal{B}(\mathbb{R}^d)} \frac{\mathbb{P}(Q(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{x})}{\mathbb{P}(Q(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{y})} \leq \mathrm{e}^{\varepsilon}.$$

Denote by $P_{Q|\boldsymbol{X}=\boldsymbol{x}}$ the conditional distribution of $Q(\boldsymbol{X}, Z)$ given $\boldsymbol{X} = \boldsymbol{x}$. We note that ε-differential privacy can be written as

$$D_\infty(P_{Q|\boldsymbol{X}=\boldsymbol{x}} \| P_{Q|\boldsymbol{X}=\boldsymbol{y}}) \leq \varepsilon\ . \tag{4.1}$$

In other words, differential privacy bounds the max-divergence distance between the conditional distributions. See Section 2.3 for comments on the distance between probability laws.

If the noise Z is independent of $\boldsymbol{X}$, then the above definition reduces to

$$\sup_{B \in \mathbb{R}^d} \frac{\mathbb{P}(Q(\boldsymbol{x}, Z) \in B)}{\mathbb{P}(Q(\boldsymbol{y}, Z) \in B)} \leq \mathrm{e}^{\varepsilon}.$$

Intuitively, failure of differential privacy means that there exists two neighbouring databases $\boldsymbol{x} \sim \boldsymbol{y}$ and an output q such that the ratio of the corresponding probabilities is large. One issue with this definition is that such q may be very unlikely, yet still the definition of differential privacy will be violated. This leads to less restrictive notion of differential privacy, as we will see below in Section 4.4.

Definition 4.2.2 (Privacy budget)**.** *The parameter $\varepsilon > 0$ is considered the privacy budget of a differentially private mechanism. It represents a limit to the amount of information about an individual in the data that can be leaked.*

The smaller ε, the more noise is added. This means that the output is more private. On the other hand, more noise also means less *data utility*.

Definition 4.2.3 (Privacy loss)**.** *Let $Q(\cdot, Z)$ be a randomized mechanism. For the given output q and given neighbouring databases $\boldsymbol{x} \sim \boldsymbol{y}$, the privacy loss is defined as*

$$\mathcal{L}_{\boldsymbol{x}\|\boldsymbol{y}}(q) = \left| \ln \frac{\mathbb{P}(Q(\boldsymbol{x}, Z) = q)}{\mathbb{P}(Q(\boldsymbol{y}, Z) = q)} \right| . \tag{4.2}$$

The above definition makes sense when for each $\boldsymbol{x}$, $Q(\boldsymbol{x}, Z)$ is a discrete random variable. Otherwise, instead of $\mathbb{P}$ in (4.2), one needs to use the density of $Q(\boldsymbol{x}, Z)$ (if it exists). This is one of the common problems faced when citing computer science literature.

4.3 Laplace noise and differential privacy

Once the definition of differential privacy is introduced, we ask the question: *What random mechanism satisfies it?* The main result of this section is the following theorem, taken from [17]. We present the proof for the sake of completeness.

Theorem 4.3.1. *Let $\varepsilon > 0$. For any $f : \mathcal{D} \to \mathbb{R}$, the randomized output perturbation mechanism*

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z$$

with the Laplace$(0, \frac{\Delta f}{\varepsilon})$*-distributed noise Z is ε-differentially private.*

Proof. Assume $\varepsilon = 1$. Recall that $g_A(q)$ is the density of a random variable A at the point q. Here, our random variable of interest is $O_f(\boldsymbol{u}, Z)$, where $\boldsymbol{u}$ is either $\boldsymbol{x}$ or $\boldsymbol{y}$. Recall that the density of Z is $g_Z(q) = \frac{1}{2\Delta f} \exp\left\{-\frac{|q|}{\Delta f}\right\}$. The privacy loss is

$$\begin{aligned} \ln \frac{g_{O_f(\boldsymbol{x},Z)}(q)}{g_{O_f(\boldsymbol{y},Z)}(q)} &= \ln \frac{g_Z(q - f(\boldsymbol{x}))}{g_Z(q - f(\boldsymbol{y}))} \\ &= \ln \frac{\exp\left\{-\frac{|q-f(\boldsymbol{x})|}{\Delta f}\right\}}{\exp\left\{-\frac{|q-f(\boldsymbol{y})|}{\Delta f}\right\}} = \frac{-|q - f(\boldsymbol{x})| + |q - f(\boldsymbol{y})|}{\Delta f} , \end{aligned}$$

and its absolute value is bounded by

$$\frac{|f(\boldsymbol{x}) - f(\boldsymbol{y})|}{\Delta f} \leq \frac{\Delta f}{\Delta f} = 1 = \varepsilon .$$

Note that the bound is uniform in $\boldsymbol{x}, \boldsymbol{y}$. Thus, the randomized output perturbation mechanism $O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z$, where Z is a random variable with a Laplace distribution, satisfies differential privacy. □

4.4 (ε, δ)-Differential Privacy

A natural question is: *Is Laplace noise the only noise mechanism that satisfies differential privacy?* From a statistical inference point of view, we want to prove the validity of differential privacy for normal noise. However, in the next example, we will show that when the noise Z is normally distributed, the similar procedure as in the example above does not lead to the uniform bound on the privacy loss.

Example 4.4.1. Consider the randomized output perturbation mechanism $O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z$, where $Z \sim \mathcal{N}(0, \sigma^2)$. The privacy loss is then,

$$\begin{aligned}
\ln \frac{g_{O_f(\boldsymbol{x},Z)}(q)}{g_{O_f(\boldsymbol{y},Z)}(q)} &= \ln \frac{\exp\left\{-\frac{1}{2}\frac{(q-f(\boldsymbol{x}))^2}{\sigma^2}\right\}}{\exp\left\{-\frac{1}{2}\frac{(q-f(\boldsymbol{y}))^2}{\sigma^2}\right\}} \\
&= -\frac{1}{2}\left(\frac{(q-f(\boldsymbol{x}))^2 - (q-f(\boldsymbol{y}))^2}{\sigma^2}\right) \\
&= -\frac{1}{2}\left(\frac{f(\boldsymbol{x})^2 - f(\boldsymbol{y})^2}{\sigma^2}\right) + \frac{q}{\sigma^2}(f(\boldsymbol{x}) - f(\boldsymbol{y})) \\
&= -\frac{1}{2}\left(\frac{(f(\boldsymbol{x}) - f(\boldsymbol{y}))(f(\boldsymbol{x}) + f(\boldsymbol{y}))}{\sigma^2}\right) + \frac{q}{\sigma^2}(f(\boldsymbol{x}) - f(\boldsymbol{y})) \\
&\leq -\frac{1}{2}\left(\frac{(f(\boldsymbol{x}) - f(\boldsymbol{y}))(f(\boldsymbol{x}) + f(\boldsymbol{y}))}{\sigma^2}\right) + \frac{q}{\sigma^2}\Delta f.
\end{aligned}$$

Note that if $q \in \mathbb{R}$, then the term $q\Delta f$ is unbounded.

The above example indicates that a normal distribution may not satisfy the definition of differential privacy. This leads to a weaker version of Differential Privacy.

Definition 4.4.2 ((ε, δ)-Differential Privacy; Approximate Differential Privacy)**.** *Let $\boldsymbol{X}$ be a database, a random element of $\mathcal{D}$. Let Z be a random element with values in a metric space $\mathcal{E}$. Let $\varepsilon > 0$ and $\delta \in (0, 1)$. A randomized mechanism $Q : \mathcal{D} \times \mathcal{E} \to \mathbb{R}^d$ is (ε, δ)-differentially private if $\forall\ \boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$, satisfying $d(\boldsymbol{x}, \boldsymbol{y}) = 1$ and all $B \in \mathcal{B}(\mathbb{R}^d)$, we have*

$$\mathbb{P}(Q(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{x}) \leq \mathrm{e}^{\varepsilon}\mathbb{P}(Q(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{y}) + \delta\ .$$

In the spirit of (4.1), we note that the (ε, δ)-differential privacy can be written as

$$D_{\infty}^{\delta}(\mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}} \| \mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{y}}) \leq \varepsilon \tag{4.3}$$

or

$$\mathbb{E}_{\exp(\varepsilon)}(\mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}} \| \mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{y}}) \leq \delta \,. \tag{4.4}$$

For the notation, see Section 2.3.

The next result shows that the Gaussian mechanism is (ε, δ)-differentially private. The result is taken from [17]. We present the proof for completeness.

Theorem 4.4.3. *Let $\varepsilon \in (0,1)$ be arbitrary and $c^2 > 2\ln(1.25/\delta)$. For any $f : \mathcal{D} \to \mathbb{R}$, the randomized output perturbation mechanism*

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z$$

with the centered Gaussian noise with the parameter $\sigma \geq c\Delta f/\varepsilon$ is (ε, δ)-differentially private.

Proof. The proof follows [17], with appropriate notational modifications.

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be neighbouring datasets, and let the query f be a real-valued function, so that $\Delta f = \Delta f_1 = \Delta f_2$. Assume $f(\boldsymbol{x}) = 0$, which implies $\Delta f = f(\boldsymbol{y})$. We examine the privacy loss ratio (4.2):

$$\ln\left(\frac{\mathbb{P}(Q(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{x})}{\mathbb{P}(Q(\boldsymbol{Y}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{y})}\right) = \left|\ln \frac{\mathrm{e}^{(-\frac{1}{2\sigma^2})q^2}}{\mathrm{e}^{(-\frac{1}{2\sigma^2})(q+\Delta f)^2}}\right| . \tag{4.5}$$

We have

$$\left|\ln \frac{e^{(-\frac{1}{2\sigma^2})q^2}}{\mathrm{e}^{(-\frac{1}{2\sigma^2})(q+\Delta f)^2}}\right| = |\ln \mathrm{e}^{(-\frac{1}{2\sigma^2})[q^2-(q+\Delta f)^2]}| = \left|\frac{q\Delta f}{\sigma^2} + \frac{\Delta^2 f}{2\sigma^2}\right|$$

When $q < \dfrac{\sigma^2\varepsilon}{\Delta f} - \dfrac{\Delta f}{2}$, the privacy loss is bounded by ε. To ensure that the privacy loss is bounded by ε with probability at least $1-\delta$, it is necessary to demonstrate that the probability of

$$\mathbb{P}\Big(|Z_\sigma| \geq \frac{\sigma^2\varepsilon}{\Delta f} - \frac{\Delta f}{2}\Big) < \delta \,,$$

where Z_σ is a centered Gaussian random variable with variance σ^2. The objective is to find the value of σ that satisfies the condition $\mathbb{P}\Big(q \geq \frac{\sigma^2\varepsilon}{\Delta f} - \frac{\Delta f}{2}\Big) < \delta/2$. Assume $\varepsilon \leq 1 \leq \Delta f$. We recall the standard tail bound, given by the probability density

function of a standard normal random variable, Z, is: $\mathbb{P}(Z > t) \leq \frac{\sigma}{\sqrt{2\pi}} e^{-t^2/2\sigma^2}$. We obtain the inequality

$$\frac{\sigma}{\sqrt{2\pi}t} e^{-t^2/2\sigma^2} < \delta/2 \iff \ln(t/\sigma) + \frac{t^2}{2\sigma^2} > \ln\left(\frac{2}{\sqrt{2\pi}\delta}\right)$$

with $t = \dfrac{\sigma^2\varepsilon}{\Delta f} - \dfrac{\Delta f}{2}$. We conclude that $c^2 > \ln(2/\pi) + 2\ln(1/\delta) + \ln(e^{8/9})$, which is satisfied when $c^2 > 2\ln(1.25/\delta)$.

Let $\mathbb{R} = R_1 \cup R_2$, where $R_1 = \{q \in \mathbb{R} : |q| \leq c\Delta f/\varepsilon\}$ and $R_2 = \{q \in \mathbb{R} : |q| > c\Delta f/\varepsilon\}$. Fix $B \subseteq \mathbb{R}$ and define $B_1 = \{f(\boldsymbol{X}) + q | q \in R_1\}$ and $B_2 = \{f(\boldsymbol{X}) + q | q \in R_2\}$.

$$\begin{aligned}
\mathbb{P}(f(\boldsymbol{X}) + q \in B \mid \boldsymbol{X} = \boldsymbol{x}) &= \mathbb{P}(f(\boldsymbol{X}) + q \in B_1 \mid \boldsymbol{X} = \boldsymbol{x}) + \mathbb{P}(f(\boldsymbol{X}) + q \in B_2 \mid \boldsymbol{X} = \boldsymbol{x}) \\
&\leq \mathbb{P}(f(\boldsymbol{X}) + q \in B_1 \mid \boldsymbol{X} = \boldsymbol{x}) + \delta \\
&\leq \mathrm{e}^{\varepsilon}\mathbb{P}(f(\boldsymbol{X}) + q \in B_1 \mid \boldsymbol{X} = \boldsymbol{y}) + \delta \ .
\end{aligned}$$

It can thus be concluded that the Gaussian mechanism is (ε, δ)-differentially private. □

4.5 Properties

In what follows, we state and prove a number of properties associated with differentially private mechanisms. Some of these properties are valid for both sanitized response (SR) and output perturbation (OP) mechanisms, while others are specific to one or the other. We present proofs for selected statements only. The majority of the results presented are taken from existing literature, with the proofs adapted to align with the mathematical framework in this thesis.

The following properties are presented for the reader's consideration:

- Preservation under different queries - valid for SR mechanisms only; see Lemma 4.5.1.
- Closure under deterministic post-processing - valid for both SR and OP mechanisms; see Lemma 4.5.4.
- Closure under independent random post-processing - valid for both SR and OP mechanisms; see Lemma 4.5.5.

4.5.1 Preservation of differential privacy under different queries

The first result indicates that if a sanitized response mechanism is ε-differentially private for a given query, then it is also ε-differentially private for any query. This is a straightforward consequence of the fact that in a S_f mechanism, noise is added to the database. The identity query, denoted by Id, is a special case.

Lemma 4.5.1 (Theorem 3.4 in [26]). *Let $\mathcal{D} = \mathbb{R}^d$. Let $\boldsymbol{Z}$ be a random vector with values in $\mathbb{R}^d$. Assume that $S_{\mathrm{Id}}(\cdot, \boldsymbol{Z})$ is (ε, δ)-differentially private. Then $S_f(\cdot, \boldsymbol{Z})$ is (ε, δ)-differentially private for any query f.*

Proof. It is sufficient to consider the case where$\delta = 0$. Let B be an arbitrary Borel set in $\mathbb{R}^d$. If $S_{\mathrm{Id}}(\cdot, \boldsymbol{Z})$ is ε-differentially private, then we can write

$$\begin{aligned}\mathbb{P}(S_{\mathrm{Id}}(\boldsymbol{x}, \boldsymbol{Z}) \in B) &\leq \mathrm{e}^{\varepsilon}\mathbb{P}(S_{\mathrm{Id}}(\boldsymbol{y}, \boldsymbol{Z}) \in B)\ ,\\ \mathbb{P}(G(\boldsymbol{x} + \boldsymbol{Z}) \in B) &\leq \mathrm{e}^{\varepsilon}\mathbb{P}(G(\boldsymbol{y} + \boldsymbol{Z}) \in B)\end{aligned}$$

with $G = \mathrm{Id}$. Let now f be an arbitrary query and $\mathbf{x} \sim \mathbf{y}$. Then we can write $S_{f\circ g} = S_f$ and

$$\begin{aligned}\mathbb{P}(S_f(\boldsymbol{x}, \boldsymbol{Z}) \in B) &= \mathbb{P}(S_{f\circ G}(\boldsymbol{x}, \boldsymbol{Z}) \in B)\\ &= \mathbb{P}(f \circ G(\boldsymbol{x}, \boldsymbol{Z}) \in B) = \mathbb{P}(G(\boldsymbol{x}, \boldsymbol{Z}) \in f^{-1}(B))\\ &\leq \mathrm{e}^{\varepsilon}\mathbb{P}(G(\boldsymbol{y}, \boldsymbol{Z}) \in f^{-1}(B)) = \mathrm{e}^{\varepsilon}\mathbb{P}(f \circ G(\boldsymbol{y}, \boldsymbol{Z}) \in B)\\ &= \mathrm{e}^{\varepsilon}\mathbb{P}(S_f(\boldsymbol{y}, \boldsymbol{Z}) \in B)\ ,\end{aligned}$$

So we conclude that if $S_{\mathrm{Id}}(\cdot, \boldsymbol{Z})$ is ε-differentially private, then $S_f(\cdot, Z)$ is ε-differentially private for any query f. □

Example 4.5.2 (Example 3.5 in [26]). For the sake of argument, consider the scenario where the same real-valued query f is asked multiple (m) times on a database $\mathcal{D} = \mathcal{U}^n$. This can be represented as a query, $f^{(m)} : \mathcal{D} \to \mathbb{R}^m$. Indeed, for $x \in \mathcal{D}$, $f^{(m)}(\boldsymbol{x}) = (f(\boldsymbol{x}), \ldots, f(\boldsymbol{x}))$. The result in Lemma 4.5.1 implies that the repeated query is also ε-differentially private. Intuitively this is clear, due to the fact that the addition of noise to the database and repeated application of *the same* query do not yield any new information.

Lemma 4.5.1 does not extend to an output perturbation mechanism. **In other words, if we have (ε, δ)-differential privacy for one query, we cannot conclude it directly for another query.** This is illustrated in the next example.

Example 4.5.3. Consider a simple binary-query, $f : \mathcal{D} \to \{0, 1\}$. Then, $\mathcal{B} = \{0, 1\}$. The output perturbation is specified by the distributions :

$$\mathbb{P}(Z_i = i) = 1 - p$$

and

$$\mathbb{P}(Z_i \neq i) = p\ ,$$

for $i = 0, 1$. It is straightforward to demonstrate that the output perturbation mechanism $O_f(\boldsymbol{x}, Z)$ is (ε, δ)-differentially private if and only if $p \geq \frac{1-\delta}{1+e^{\varepsilon}}$. To illustrate this example,

we assume that there exists two databases $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$ for which $f(\boldsymbol{x}) = 0$ and $f(\boldsymbol{y}) = 1$. Consider the set $B = \{0\}$, then we can explicitly calculate the probabilities

$$\mathbb{P}(O_f(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{x}) = 1 - p$$

and

$$\mathbb{P}(O_f(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{y}) = p \, .$$

Let $\varepsilon = \ln(3), \delta = 0.1$, and $p = \frac{1}{4}$. We first query one time to show that differential privacy holds:

$$\begin{aligned} \mathbb{P}(O_f(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{x}) &\leq \mathrm{e}^{\varepsilon} \cdot \mathbb{P}(O_f(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{y}) + \delta \\ 1 - p &\leq \mathrm{e}^{\varepsilon} \cdot p + \delta \\ \frac{3}{4} &\leq \frac{3}{4} + 0.1. \end{aligned}$$

Thus, (ε, δ)-differential privacy holds when the database is queried on a single occasion. A second query of the database is then conducted to ascertain whether differential privacy still holds. We want to verify the inequality:

$$\begin{aligned} &\mathbb{P}((O_f(\boldsymbol{X}, Z), O_f(\boldsymbol{X}, Z)) \in B \times B \mid \boldsymbol{X} = \boldsymbol{x}) \\ &\leq \mathrm{e}^{\varepsilon} \cdot \mathbb{P}((O_f(\boldsymbol{X}, Z), O_f(\boldsymbol{X}, Z)) \in B \times B \mid \boldsymbol{X} = \boldsymbol{y}) + \delta \, . \end{aligned}$$

The left-hand side is given by the expression $(1-p)^2$, while the right-hand side becomes $\leq \mathrm{e}^{\varepsilon} \cdot p^2 + \delta$. That is, $\frac{9}{16} \not\leq \frac{3}{16} + 0.1$.

It can thus be concluded that if the output perturbation mechanism is applied twice in this scenario, it will fail to preserve differential privacy.

4.5.2 Post-processing

In the next result, a randomized mechanism Q is either SR (Sanitized Response) or OP (Output Perturbation).

Lemma 4.5.4 (Post-processing)**.** *Assume that $Q(\cdot, Z)$ is a ε-differentially private response mechanism with values in $\mathbb{R}^d$. Let $g : \mathbb{R}^d \to \mathbb{R}^{d'}$ be a deterministic function. Then $g \circ Q(\cdot, Z)$ is ε-differentially private.*

Let f be a $\mathbb{R}^d$-valued function. If O_f is the output perturbation mechanism and $g(z, z') = z + z'$, where $z, z' \in \mathbb{R}^d$, then $g \circ O_f$ becomes

$$g \circ O_f(\boldsymbol{x}, z, z') = f(\boldsymbol{x}) + z + z' \, ,$$

and it maps $\mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d$ into $\mathbb{R}^d$. According to Lemma 4.5.4, $g \circ O_f(\cdot, Z, z')$ is ε-differentially private for any z', whenever $O_f(\cdot, Z)$ is ε-differentially private. Recall that

$\boldsymbol{x}$ plays a role of the database and the dimension of the noise z is related to the dimension of the query, not the database (hence the notation z for the noise).

The same reasoning is applied to the sanitized response mechanism S_f and

$$g \circ S_f(\boldsymbol{x}, \boldsymbol{z}, z') = f(\boldsymbol{x} + \boldsymbol{z}) + z' \ .$$

Here the noise $\boldsymbol{z}$ has the same dimension as the database, while z' has the dimension of the query f.

The next lemma demonstrated that differential privacy is preserved when g is considered as a random map.

Lemma 4.5.5. *Let $\boldsymbol{Z}$ and $\boldsymbol{Z}'$ be two independent random vectors with values in $\mathbb{R}^d$. Assume that $Q_f(\cdot, Z)$ is ε-differentially private. Then $g \circ Q_f(\cdot, Z, Z')$ is also ε-differentially private.*

Proof. Denote $H = g \circ O_f$. Then $H(\boldsymbol{x}, Z, Z') = f(\boldsymbol{x}) + Z + Z'$. Let $F_{Z'}$ be the distribution function of Z'. Then by the principle of independence and the differential privacy property of $O_f(\cdot, Z)$, we have

$$\begin{aligned}
&\mathbb{P}(H(\boldsymbol{x}, Z, Z') \in B) = \int \mathbb{P}(H(\boldsymbol{x}, Z, z') \in B) F_{Z'}(dz') \\
&= \int \mathbb{P}(f(\boldsymbol{x}) + Z \in B - z') F_{Z'}(dz') \leq \mathrm{e}^{\varepsilon} \int \mathbb{P}(f(\boldsymbol{y}) + Z \in B - z') F_{Z'}(dz') \\
&= \int \mathbb{P}(H(\boldsymbol{y}, Z, z') \in B) F_{Z'}(dz') = \mathrm{e}^{\varepsilon} \mathbb{P}(H(\boldsymbol{y}, Z, Z') \in B).
\end{aligned}$$

The proof for $g \circ S_f$ is analogous. □

In the aforementioned proof, the crucial aspect is the independence between Z and Z'. In the following example, we demonstrate that if the independence assumption is relaxed, the conclusion of Lemma 4.5.5 is no longer valid. One might observe an apparent contradiction with the statement made after Proposition 2.1 in [17]. Nevertheless, it seems reasonable to posit that the authors work under the assumption of independence. In short, differential privacy is not preserved under dependent post-processing.

Example 4.5.6. Let f be a real-valued function and define $g \circ O_f(\boldsymbol{x}, z, z') = f(\boldsymbol{x}) + z + z'$, $z, z' \in \mathbb{R}$. Assume that Z is a Laplace random variable. Then set $Z' = -Z + \mathcal{N}$, where $\mathcal{N}$ is a Normal random variable. Then, it can be shown that $O_f(\boldsymbol{x}, Z)$ is ε-differentially private. However, since $g \circ O_f(\boldsymbol{x}, Z, Z') = f(\boldsymbol{x}) + \mathcal{N}$, then $g \circ O_f(\cdot, Z, Z')$ is not differentially private.

We make another important observation in relation to Lemma 4.5.5: the addition of further noise, Z, does not result in an increase in the level of privacy, ε. This is not a particularly intuitive result and, in fact, represents a significant fallacy of differential privacy.

Example 4.5.7 (Sum of two independent Laplace noises)**.** Assume that the following random variables Z and Z' are independent with a density Laplace$(0,1)$. Then, $\xi = Z + Z'$ has a density given by

$$f_\xi(q) = \frac{1}{4}(1+|q|)\mathrm{e}^{-|q|} \ , \quad q \in \mathbb{R} \ .$$

Let $O_f(\boldsymbol{x}, \xi) = f(\boldsymbol{x}) + \xi$. The ratio of the densities of $f(\boldsymbol{x}) + \xi$ and $f(\boldsymbol{y}) + \xi$ is

$$\frac{(1+|q-f(\boldsymbol{x})|)\mathrm{e}^{-|q-f(\boldsymbol{x})|}}{(1+|q-f(\boldsymbol{y})|)\mathrm{e}^{-|q-f(\boldsymbol{y})|}} \ .$$

Assume that $f(\boldsymbol{x}) \neq f(\boldsymbol{y})$. We see that when $q \to f(\boldsymbol{y})$, the expression above is unbounded. Indeed the ratio behaves like

$$\lim_{x\to\infty} \frac{e^x}{1+x} = \infty \ .$$

Hence, we conclude that the ratio of densities is unbounded when $q \to f(\boldsymbol{y})$. This serves to illustrate that when the output perturbation mechanism is considered with the noise being the sum of two differentially private Laplace random variables, the result is not differentially private.

The next examples illustrate the lack of impact of adding additional noise.

Example 4.5.8. Assume that Z, Z' are independent random variables with the densities Laplace$(\frac{\Delta f}{\varepsilon})$, Laplace$(\frac{\Delta f}{\varepsilon'})$ respectively. Then the randomized output perturbation mechanism $f(\boldsymbol{x}) + Z + Z'$ is $(\varepsilon \wedge \varepsilon')$-differentially private. Indeed, this is the optimal achievable result. The primary reason for this is that the combination of the two noise variables $Z + Z'$ does not produce a differentially private noise. See Example 4.5.7. Indeed, similar to Lemma 4.5.5, denote $H = g \circ O_f$. Then $H(\boldsymbol{x}, Z, Z') = f(\boldsymbol{x}) + Z + Z'$. Let F_Z,$F_{Z'}$ be the distribution functions of Z and Z' respectively. We condition first on Z' and obtain

$$\begin{aligned}
&\mathbb{P}\left(f(\boldsymbol{x}) + Z + Z' \in B\right) = \int \mathbb{P}(f(\boldsymbol{x}) + Z + z' \in B) F_{Z'}(dz') \\
&= \int \mathbb{P}(f(\boldsymbol{x}) + Z \in B - z') F_{Z'}(dz') \\
&\leq \mathrm{e}^{\varepsilon} \int \mathbb{P}(f(\boldsymbol{y}) + Z \in B - z') F_{Z'}(dz') \\
&= \mathrm{e}^{\varepsilon} \mathbb{P}\left(f(\boldsymbol{y}) + Z + Z'\right) \in B) \, .
\end{aligned}$$

Similarly, if we condition on Z we obtain

$$\begin{aligned}
&\mathbb{P}\left(f(\boldsymbol{x})+Z+Z' \in B\right) = \int \mathbb{P}(f(\boldsymbol{x})+z+Z' \in B) F_Z(dz) \\
&= \int \mathbb{P}(f(\boldsymbol{x})+Z' \in B-z) F_Z(dz) \\
&\leq \mathrm{e}^{\varepsilon'} \int \mathbb{P}(f(\boldsymbol{y})+Z' \in B-z) F_Z(dz) \\
&= \mathrm{e}^{\varepsilon'} \mathbb{P}\left(f(\boldsymbol{y})+Z+Z' \in B\right).
\end{aligned}$$

So we conclude that the output perturbation mechanism, $O_f(\boldsymbol{x}, Z, Z') = f(\boldsymbol{x}) + Z + Z'$ is $(\varepsilon \wedge \varepsilon')$- differentially private.

Example 4.5.9 (Sum of independent Laplace and normal noises)**.** Let $W = X + Y$ where $X \sim \mathcal{N}(\mu, \sigma^2)$, with density $f_X(x; \mu, \sigma) = \phi((x-\mu)/\sigma)/\sigma$ (ϕ is the standard normal density)) and $Y \sim$ Exponential(1) with density $f_Y(y) = e^{-y}\mathcal{I}(y > 0)$.
We can write the convolution as

$$f_W(w; \mu, \sigma) = \int_{\infty}^{\infty} f_Y(y) f_X(w-y; \mu, \sigma) dy = \int_0^{\infty} e^{-y} f_X(w-y; \mu, \sigma) dy \,.$$

Substituting $\sigma z = w - y - \mu$, we get

$$f_W(w; \mu, \sigma) = e^{\mu - w + \sigma^2/2} \int_{-\infty}^{(w-\mu)/\sigma} \phi(z - \sigma) dz = e^{\mu - w + \sigma^2/2} \Phi\left(\frac{w-\mu}{\sigma} - \sigma\right) \,,$$

where Φ is the standard normal CDF. A symmetric Laplace random variable U can be expressed as a sum of a "positive" scaled exponential random variable and a "negative" scaled exponential random variable:

$$U = U_- + U_+,$$

where $U_+ = \beta Y$ and $U_- = -\beta Y$, with positive scale β. Adding X, the two components can be written as

$$W_+ = U_+ + X = \beta\left(Y + \left(\frac{\sigma}{\beta} Z + \frac{\mu}{\beta}\right)\right) \tag{4.6}$$

and

$$W_- = U_- + X = -\beta\left(Y + \left(-\frac{\sigma}{\beta} Z + \frac{\mu}{\beta}\right)\right) \,, \tag{4.7}$$

where Z is standard normal. Noting that Z and $-Z$ have the same distribution, we can rewrite (4.6) and (4.7) as

$$f_{W_+}(w; \mu, \sigma, \beta) = \frac{1}{\beta} f_W\left(\frac{w}{\beta}; \frac{\mu}{\beta}, \frac{\sigma}{\beta}\right) \,,$$

$$f_{W_-}(w;\mu,\sigma,\beta) = \frac{1}{\beta} f_W\left(-\frac{w}{\beta}; -\frac{\mu}{\beta}, \frac{\sigma}{\beta}\right) ,$$

and thus

$$f_W(w;\mu,\sigma,\lambda,\beta,p) = \frac{1}{2} f_{W_+}(w;\mu,\sigma,\lambda,\beta) + \frac{1}{2} f_{W_-}(w;\mu,\sigma,\lambda,\beta) . \tag{4.8}$$

A plot is presented that compares a histogram of one million independent and identically distributed draws from a symmetric Normal-Laplace distribution with parameters $\mu = 0$, $\sigma = 1$, and $\beta = 2$ to a calculation based directly on the formulas (4.6), (4.7), and (4.8).

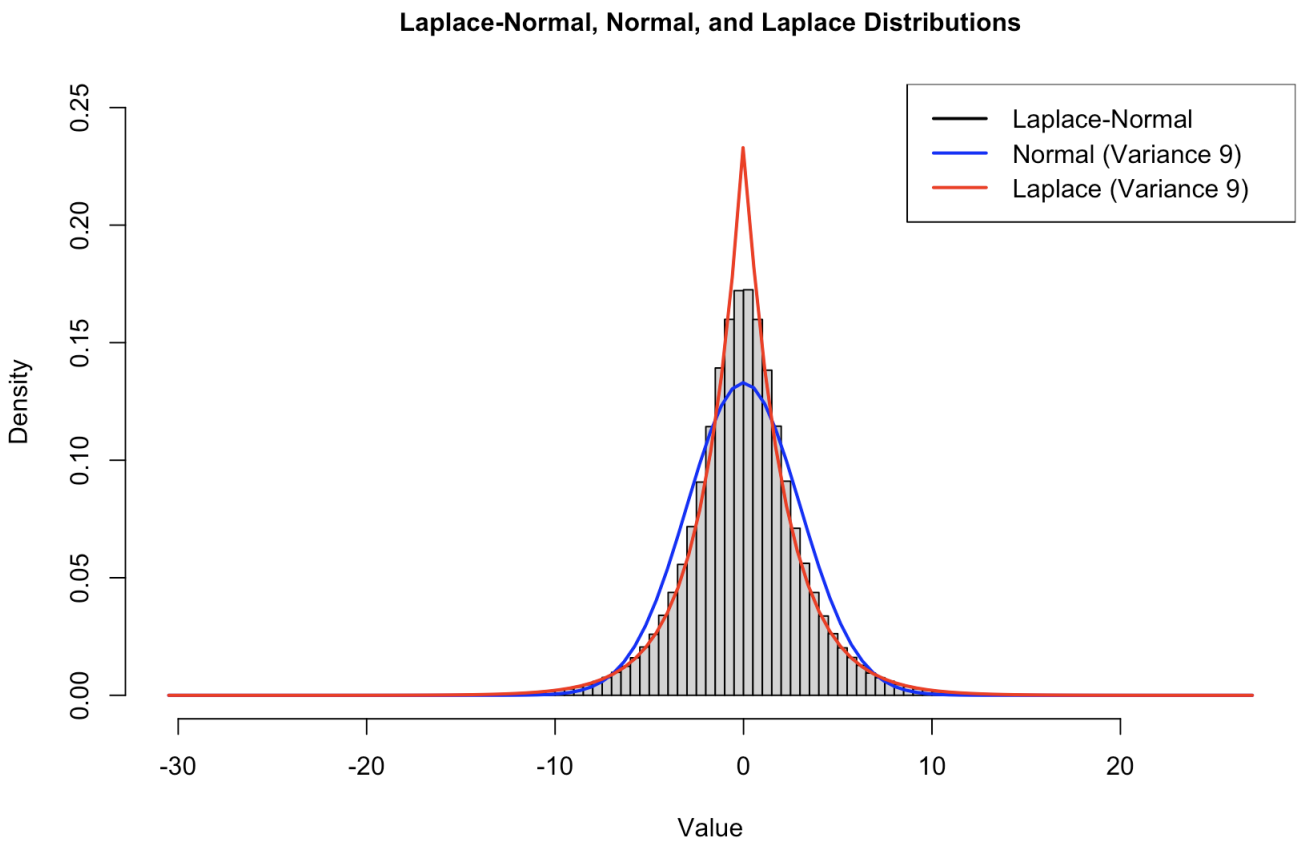

Figure 4.1: Histogram of Laplace-Normal density
Histogram of i.i.d. samples from a symmetrical Normal-Laplace distribution with Normal and Laplace densities with a variance of 9.

The above discussion leads to the conclusion that,

DP Fallacy 4.5.10. *Adding additional noise does not improve differential privacy.*

4.5.3 Group privacy

Recall that the definition of differential privacy assumed that $\boldsymbol{x} \sim \boldsymbol{y}$. That is the Hamming distance is one: $d(\boldsymbol{x}, \boldsymbol{y}) = 1$. In other words, two databases differ by one record. The definition can be extended to a situation when two databases differ by more than one record.

Lemma 4.5.11 (Group privacy)**.** *Let f be a real-valued function and Z be a random variable. Assume that a randomized mechanism $Q(\cdot, Z)$ is ε-differentially private. Let k be a positive integer. Then*

$$\sup_{\boldsymbol{x},\boldsymbol{y}:d(\boldsymbol{x},\boldsymbol{y})=k} \sup_{B\in\mathcal{B}} \frac{\mathbb{P}(Q(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{x})}{\mathbb{P}(Q(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{y})} \leq \mathrm{e}^{k\varepsilon}.$$

Proof. We use the mechanism $O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z$, where $Z \sim \text{Laplace}(\Delta f/\varepsilon)$. We compare the ratio of densities at an arbitrary point z and examine what differential privacy bound we can achieve. We note that we want to compare $\boldsymbol{x} = (x_1, \cdots, x_n)$ and $\boldsymbol{y} = (y_1, \cdots, y_n)$. We denote $\boldsymbol{x}_{(-i)} = (x_{i+1}, \cdots, x_n)$ and $\boldsymbol{y}_{(i)} = (y_1, \cdots, y_i, x_{(-i)})$. It is obvious that $d(\boldsymbol{x}, \boldsymbol{y}) = n$ and applying repeatedly the definition of ε-differential privacy,

$$\begin{aligned}
\mathbb{P}(f(\boldsymbol{x}) + Z \in B) &\leq \mathrm{e}^{\varepsilon}\mathbb{P}(f(\boldsymbol{y}_{(1)}) + Z \in B) \\
&\leq \mathrm{e}^{\varepsilon}\mathrm{e}^{\varepsilon}\mathbb{P}(f(\boldsymbol{y}_{(2)}) + Z \in B) \\
&\leq \mathrm{e}^{n\varepsilon}\mathbb{P}(f(\boldsymbol{y}) + Z \in B)
\end{aligned}$$

as required. □

Lemma 4.5.12 (Group privacy for (ε, δ)-differential privacy)**.** *Assume that a randomized mechanism $Q(\cdot, Z)$ is (ε, δ)-differentially private. Let k be a positive integer. Then*

$$\sup_{\boldsymbol{x},\boldsymbol{y}:d(\boldsymbol{x},\boldsymbol{y})=k} \sup_{B\in\mathcal{B}} \mathbb{P}(Q(\boldsymbol{x}, Z) \in B) \leq \mathrm{e}^{k\varepsilon}\mathbb{P}(Q(\boldsymbol{y}, Z) \in B) + \delta^{(k)},$$

with $\delta^{(k)} = \delta \sum_{j=0}^{k-1} \mathrm{e}^{j\varepsilon}$.

Proof. We use the same notation as in Lemma 4.5.11. Then

$$\begin{aligned}
\mathbb{P}(f(\boldsymbol{x}) + Z \in B) &\leq \mathrm{e}^{\varepsilon}\mathbb{P}(f(\boldsymbol{y}_{(1)}) + Z \in B) + \delta \\
&\leq \mathrm{e}^{\varepsilon}\left(\mathrm{e}^{\varepsilon}\mathbb{P}(f(\boldsymbol{y}_{(2)}) + Z \in B) + \delta\right) + \delta \\
&\leq \mathrm{e}^{k\varepsilon}\mathbb{P}(f(\boldsymbol{y}) + Z \in B) + \sum_{j=0}^{k-1} \mathrm{e}^{j\varepsilon}\delta =: \mathrm{e}^{k\varepsilon}\mathbb{P}(f(\boldsymbol{y}) + Z \in B) + \delta^{(k)}.
\end{aligned}$$

□

4.5.4 Compositions

Let $Q^{(i)} : \mathcal{D} \times \mathcal{N} \to \mathbb{R}^{d^{(i)}}$, $i = 1, \dots, m$, be response mechanisms. Define $Q : \mathcal{D} \times \mathcal{N}^m \to \mathbb{R}^{\sum_{i=1}^m d^{(i)}}$ by

$$Q(\boldsymbol{x}, (\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)})) = (Q^{(1)}(\boldsymbol{x}, \mathbf{z}^{(1)}), \dots, Q^{(m)}(\boldsymbol{x}, \mathbf{z}^{(m)})) \ .$$

Therefore, we apply multiple queries to the **same** database. As such, the privacy deteriorates.

Lemma 4.5.13. *Let $Z^{(i)}$, $i = 1, \dots, m$, be independent random elements. If $Q^{(i)}(\cdot, Z^{(i)})$, $i = 1, \dots, m$, are $\varepsilon^{(i)}$-differentially private, then $Q(\cdot, (Z^{(1)}, \dots, Z^{(m)}))$ is $\sum_{i=1}^m \varepsilon^{(i)}$-differentially private.*

Proof. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$ be such that $d(\boldsymbol{x}, \boldsymbol{y}) = 1$, and consider any outcome $B \times \cdots \times B \in \mathcal{B}$. Then

$$\begin{aligned} \frac{\mathbb{P}(Q(\boldsymbol{x}, (Z^{(1)}, \cdots, Z^{(m)})) \in B \times \cdots \times B)}{\mathbb{P}(Q(\boldsymbol{y}, (Z^{(1)}, \cdots, Z^{(m)})) \in B \times \cdots \times B)} &= \prod_{i=1}^m \frac{\mathbb{P}(Q(\boldsymbol{x}, Z^{(i)} \in B)}{\mathbb{P}(Q(\boldsymbol{y}, Z^{(i)} \in B)} \\ &\leq e^{\varepsilon^{(1)}} \cdots e^{\varepsilon^{(m)}} \\ &= e^{\sum_{i=1}^m \varepsilon^{(i)}} \end{aligned}$$

□

Assume now that the database $\boldsymbol{x}$ is decomposed into **disjoint** databases $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$. Let $Q^{(i)} : \mathcal{D} \times \mathcal{N} \to \mathbb{R}^{d^{(i)}}$, $i = 1, \dots, m$, be response mechanisms. Define $Q : \mathcal{D} \times \mathcal{N}^m \to \mathbb{R}^{\sum_{i=1}^m d^{(i)}}$ by

$$Q(\boldsymbol{x}, (z^{(1)}, \dots, z^{(m)})) = (Q^{(1)}(\boldsymbol{x}_1, z^{(1)}), \dots, Q^{(m)}(\boldsymbol{x}_m, z^{(m)})) \ . \tag{4.9}$$

Therefore, we apply multiple queries to **disjoint** databases. As such, the privacy does not deteriorate.

Lemma 4.5.14. *Let $Z^{(i)}$, $i = 1, \dots, m$, be independent random elements. If $Q^{(i)}(\cdot, Z^{(i)})$, $i = 1, \dots, m$, are $\varepsilon^{(i)}$-differentially private, then the query (4.9) is $\max_{i=1,\dots,m} \varepsilon^{(i)}$-differentially private.*

Proof. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$ be such that $d(\boldsymbol{x}, \boldsymbol{y}) = 1$. If $\boldsymbol{x}$ and $\boldsymbol{y}$ are decomposed into $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$ and $\boldsymbol{y}_1, \dots, \boldsymbol{y}_m$, respectively, then only one of the m pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, $i = 1, \dots, m$, can differ by one record, say $i = 1$. Consider any outcome $B \times \cdots \times B \in \mathcal{B}$. Then

$$\begin{aligned} \frac{\mathbb{P}(Q(\boldsymbol{x}, (Z^{(1)}, \cdots, Z^{(m)})) \in B \times \cdots \times B)}{\mathbb{P}(Q(\boldsymbol{y}, (Z^{(1)}, \cdots, Z^{(m)})) \in B \times \cdots \times B)} &= \prod_{i=1}^{(m)} \frac{\mathbb{P}(Q(\boldsymbol{x}_i, Z^{(i)} \in B)}{\mathbb{P}(Q(\boldsymbol{y}_i, Z^{(i)} \in B)} = \frac{\mathbb{P}(Q(\boldsymbol{x}_1, Z^{(1)} \in B)}{\mathbb{P}(Q(\boldsymbol{y}_1, Z^{(1)} \in B)} \\ &\leq e^{\varepsilon^{(1)}} \ . \end{aligned}$$

□

Chapter 5

Differential Privacy from a data utility perspective

5.1 Introduction

As stated in Chapter 4, "Differential Privacy is a probabilistic guarantee that the inclusion of an individual in the database does not alter the outcome of a query on the database by more than a specific bound." This is achieved by adding a noise to a query. The amount of noise is determined by the **global** sensitivity of the query (see Definition 2.1.2). However, this can be highly detrimental from the standpoint of data utility.

To be more precise, consider a database $\boldsymbol{x} = (x_1, \ldots, x_n)$ derived from a population $\mathcal{P}$. We can consider two distinct scenarios for a database. It may be treated as fixed, in which case we will denote it by $\boldsymbol{x}$. Alternatively, it may be treated as random, in which case it will be written as $\boldsymbol{X}$. If the objective is to estimate a population parameter θ, we can use a point estimator $f(\boldsymbol{x})$ for a suitable function f. For example, if θ represents the population mean, then $f(\boldsymbol{x})$ is the sample mean, $\bar{\boldsymbol{x}}$. In the context of privacy, we consider the randomized output perturbation mechanism, defined as follows:

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z \ ,$$

where Z is a random variable (or a random vector if f has values in $\mathbb{R}^d$). In other words, the estimator of θ is perturbed by a random noise. The additional noise Z contributes to a decline in the utility of the data. For example, if $Z \sim \text{Lap}(\Delta f/\varepsilon)$, we can compare the mean square error (MSE) of both the original and privatized estimators. The MSE of the estimator θ is given by:

$$\text{MSE}(f(\boldsymbol{X})) = \mathbb{E}[(f(\boldsymbol{X}) - \theta)^2] \ ,$$

$$\text{MSE}(O_f(\boldsymbol{X}, Z)) = \text{MSE}(f(\boldsymbol{X})) + \text{Var}(Z) = \mathbb{E}[(f(\boldsymbol{X}) - \theta)^2] + 2(\Delta f/\varepsilon)^2 \ .$$

The term $2(\Delta f/\varepsilon)^2$ describes the additional contribution from the privatization, regardless of whether the database is treated as random or not.

Above, Δf is the **global** sensitivity of the query f. If $f(\boldsymbol{x})$ is the sample mean and the population has a range of $[0, \Lambda]$, then the sensitivity is given by $\Delta f = \frac{\Lambda}{n}$ (cf. Example 2.1.3). Consequently, the variance of the noise is given by:

$$\mathrm{Var}(Z) = 2\left(\frac{\Lambda^2}{n^2\varepsilon^2}\right) . \tag{5.1}$$

As a result, for large values of Λ, a considerable amount of noise is introduced. Additionally, the range may be either unknown or infinite. The situation becomes even more challenging when a normal noise is introduced, as will be discussed below. Consequently, one of the most important practical issues is to **reduce the amount of noise added while maintaining the same level of privacy.**

An alternative approach to **global** sensitivity is to use the **local** sensitivity (see Definition 2.1.1), which is based on the database itself. However,

DP Fallacy 5.1.1. *The local sensitivity may lead to a violation of Differential Privacy.*

An alternative approach is to use an "intermediate" version between the local and global sensitivity. This is referred to as **smooth sensitivity**, originally introduced in [39]. However, it should be noted that some results in the latter reference are stated incorrectly.

These challenges, including an example of the violation of differential privacy in this context, are discussed in Section 5.2. In Section 5.2.2, we present a general approach to controlling "sensitivity", which is based on the ideas of smooth sensitivity. This approach has many advantages from the perspective of data utility. The contents of Section 5.2.2 is an original work of the author of the thesis. The approach presented in that section is being used later in Section 5.4.

The classical definition of differential privacy allows for the use of only a Laplace noise. From a statistical inference perspective, it may be more appropriate to use a normal noise. It should be noted that while a Gaussian distribution violates differential privacy, it aligns with the framework for approximate differential privacy. The Gaussian output perturbation mechanism satisfies the (ε, δ)-differentially private whenever the variance $\sigma^2 > 2\Delta f \ln(1.25/\delta)/\varepsilon$ (cf. Theorem 4.4.3). The parameter δ must be relatively small, which implies that the variance must be large. Consequently, a considerable amount of noise must be added to achieve the desired level of privacy. This has an unsurprisingly negative effect on data utility. In short,

DP Fallacy 5.1.2. *Differential Privacy is not well suited to a normal distribution.*

This issue is addressed in detail in Section 5.3. We propose the **Mixed Noise Mechanism** (MNM) that fulfills the requirements for approximate differential privacy and allows for the addition of Gaussian noise with a well controlled variance when the sensitivity is low. This is a common occurrence in many real-world applications. The MNM mechanism is superior to the classical Laplace mechanism in terms of data utility. The content of that section is based on the author's original work, [8].

Still in the race to decrease the amount of noise added, we recall the well-known fact that averaging serve to decreases variability. Let us consider the following simple example: Assume that we have a database with real-valued entries. For the sake of illustration, consider two scenarios. In the first scenario, the population is uniformly distributed over the interval $[0, \Lambda]$ with $0 < \Lambda < \infty$. In the second scenario, the population exhibits a skewed distribution on the interval $[0, \Lambda]$, with a small probability of values exceeding $\Lambda/2$. Consider the output perturbation mechanism with the mean query. In both scenarios, the global sensitivity is proportional to Λ. A random sample from the second population will likely exhibit a reduced prevalence of "outliers" (values approaching the upper limit of the interval) in comparison to a sample drawn from the first population. It is therefore necessary to protect these few outliers in the same way as all the other "average" observations. This gives rise to the following issue:

DP Fallacy 5.1.3. *All entries in the database are treated in the same manner.*

Another fallacy that is worthy of mention is as follows. If we multiply each entry in the dataset by $a > 1$, and then we apply a query f to the dataset, we must add additional noise to achieve the same level of privacy.

DP Fallacy 5.1.4. *Scaling each entry in the database by $a > 1$ leads to more noise being added.*

These issues result in the addition of excessive noise to the query, which ultimately diminishes the utility of the data. In Section 5.4 we propose two blocking algorithms that reduce the amount of noise added by exploiting the averaging principle. Furthermore, the first presented algorithm *adapts* to data variability, which leads to an improvement in data utility. The content of Section 5.4 is the author's original work.

We continue with issues related to differential privacy. Consider a sanitized response mechanism, (2.3), in conjunction with the identity query. That is, we add a noise to a database. In practice, a common problem encountered is:

DP Fallacy 5.1.5. *Adding noise with an unbounded support may lead to unrealistic data entries.*

For example, if the variable of interest is the "age" of an individual, the addition of Laplace or Normal noise may necessitate the introduction of negative values. Similarly, consider the output perturbation mechanism with the sample variance as a query.

The introduction of noise may result in the generation of a negative noisy variance. At present, there is no satisfactory approach to this issue. In Section 5.5, we present a solution based on a bounded Laplace mechanism. This approach is based on the paper [27].

Moreover, differential privacy has recently become synonymous with noise addition. From the standpoint of data utility, there is a significant distinction between the pre- and post-processing scenarios, namely the use of a sanitized response mechanism versus an output perturbation mechanism. It can be reasonably assumed that pre-processing provides a greater degree of privacy protection against all potential queries and unknown uses of the data, whereas post-processing offers a more targeted level of protection against a specific query. Therefore, if the level of privacy is fixed, it can be argued that post-processing provides greater data utility. This is discussed in detail in Section 5.6. The contents of this section is based on the author's original work.

Privacy algorithms entail the perturbation of data, for example by adding noise according to some distribution. This has implications for the utility of the data and, consequently, for statistical inference. To illustrate, if a researcher has access to a database, one can estimate the population mean by the sample mean. Furthermore, it is possible to provide confidence intervals for the population mean by applying the central limit theorem and estimating the variance (using the sample variance).

In the event that the researcher is only able to access the results of a randomized query, two issues emerge. Firstly, the researcher will estimate the population mean by a noisy sample mean, that is to say, a sample mean with noise added. The researcher may also lack access to the sample variance, unless an additional query is permitted. This presents a challenge when calculating confidence intervals or performing hypothesis testing. This is discussed in greater detail in Section 5.7 (based on the original work of the author).

Finally, an alternative approach to adapting differential privacy to a normal noise is to change the distance between distributions. Indeed, as indicated in (4.3)-(4.4), (approximate) differential privacy measures a particular distance between the conditional distributions. It is argued in [11] that changing the distance may lead to an improvement when applied to a normal distribution. We argue in Section 5.8 that modifying the definition of privacy alone is an ineffective approach to enhancing the efficacy of data protection.

5.2 Dealing with sensitivity

As previously stated at the outset of this chapter, utilizing the global sensitivity may not be a practical option. Conversely, the local sensitivity may be employed, which is readily

computable for each query and each data set (see R Code A.0.1). Nevertheless, the utilization of the local sensitivity may violate differential privacy. The following example is provided for illustrative purposes.

Example 5.2.1 (The local sensitivity is not differentially private)**.** Assume that the database is real-valued and the entries x_i come from a distribution with the support $[0, \Lambda]$, $0 < \Lambda < \infty$. Assume that n is odd. Let $f(\boldsymbol{x}) = \text{median}(x_1, \ldots, x_n)$. Let $m = \frac{n+1}{2}$ be the rank of the median element. The global sensitivity of the median is then given by $\Delta f = \Lambda$; see Example 2.1.3. Indeed, $f(x_1, \ldots, x_n) = 0$ while $f(x_1, \ldots, x_{m-1}, \Lambda, x_{m+1}, \ldots, x_n) = \Lambda$. The addition of noise in accordance with this sensitivity will result in the destruction of data utility.

By contrast, the local sensitivity is given by $\Delta^{(\text{local})} f(\boldsymbol{x}) = \max(x_m - x_{m-1}, x_{m+1} - x_m)$. This approach yields less noise, but may result in a violation of $(\varepsilon, \delta)-$differentially privacy. Indeed, consider two data sets.

- Dataset 1: $x_1 = \cdots = x_m = 0$ and $x_{m+1} = \cdots = x_n = \Lambda$;
- Dataset 2: $x_1 = \cdots = x_{m+1} = 0$ and $x_{m+2} = \cdots = x_n = \Lambda$.

It can be noted that, in both datasets, the median is equal to zero. In the first data set, the local sensitivity is also equal to Λ, whereas in the second dataset, the local sensitivity is zero. Additionally, the Hamming distance between these two datasets is 1. Thus, if the mechanism $f(\boldsymbol{x}) + Z = 0 + Z$, with Z representing a Laplace noise with a parameter proportional to the local sensitivity, then in the second scenario, no noise will be added. Therefore, the probability of receiving a non-zero response to the randomized query is zero for the second data set and non-zero for the first data set. As a result, the protocol does not satisfy the requirements to be considered $(\varepsilon, \delta)-$ differentially private.

There are no satisfactory solutions to this problem. One potential solution is obtained via the smooth sensitivity.

5.2.1 Smooth sensitivity

One potential solution to the aforementioned issue is the approach based on the *smooth sensitivity*, proposed in [39]. The concept of smooth sensitivity lies between the local and the global sensitivity. As in the case of local sensitivity, it can be computed for the given database and the query. As in the case of the global sensitivity, it yields an appropriate version of differential privacy. The findings related to the smooth sensitivity are summarized at the end of this section.

Recall that a database $\boldsymbol{x}$ has values in $\mathcal{D}$. Recall the notation $\Delta^{(\text{local})} f(\boldsymbol{x})$ for the local sensitivity of a query f; see Definition 2.1.1. Recall also that the global sensitivity is $\Delta f = \max_{\boldsymbol{x} \in \mathcal{D}} \Delta^{(\text{local})} f(\boldsymbol{x})$; see (2.2). We always have $\Delta^{(\text{local})} f(\boldsymbol{x}) \leq \Delta f$.

Definition 5.2.2 (A Smooth Bound on Local Sensitivity)**.** *Let $\beta > 0$. A function $S : \mathcal{D} \to \mathbb{R}^+$ is a β-smooth upper bound on the local sensitivity of f if it satisfies the following requirements:*

$$\begin{aligned} \forall \boldsymbol{x} \in \mathcal{D}: \quad & S(\boldsymbol{x}) \geq \Delta^{(\text{local})} f(\boldsymbol{x}); \\ \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}, d(\boldsymbol{x}, \boldsymbol{y}) = 1: \quad & S(\boldsymbol{x}) \leq e^{\beta} S(\boldsymbol{y}). \end{aligned}$$

The last property yields the following result for any two neighbouring databases:

$$e^{-\beta} \leq \frac{S(\boldsymbol{x})}{S(\boldsymbol{y})} \leq e^{\beta} \ .$$

The constant function $S(\boldsymbol{x}) = \Delta f$ (the global sensitivity), represents the 0-smooth upper bound. On the other hand, when $\beta > 0$, the function S serves as a highly conservative upper bound on the local sensitivity of f.

Definition 5.2.3 (Smooth Sensitivity)**.** *For $\beta > 0$, the β-smooth sensitivity of f is*

$$S^*_{f,\beta}(\boldsymbol{x}) = \max_{\boldsymbol{y} \in \mathcal{D}} \left(\Delta^{(local)} f(\boldsymbol{y}) \cdot e^{-\beta d(\boldsymbol{x}, \boldsymbol{y})} \right) \ . \tag{5.2}$$

The following lemma demonstrates that the function $S^*_{f,\beta}$ fulfills the conditions in Definition 5.2.2. It can be shown that this function represents the optimal β-smooth upper bound.

Lemma 5.2.4. *$S^*_{f,\beta}$ is a β-smooth upper bound on the local sensitivity. Additionally, $S^*_{f,\beta}(\boldsymbol{x}) \leq S(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{D}$ for every β-smooth upper bound S on the local sensitivity $\Delta^{(\text{local})} f(\boldsymbol{x})$.*

Proof. To show that $S^*_{f,\beta}$ is an upper bound on the local sensitivity, we can first see that

$$\begin{aligned} S^*_{f,\beta}(\boldsymbol{x}) &= \max \left(\Delta^{(\text{local})} f(\boldsymbol{x}), \max_{\boldsymbol{y} \neq \boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}} \left(\Delta^{(\text{local})} f(\boldsymbol{y}) \cdot e^{-\beta d(\boldsymbol{x}, \boldsymbol{y})} \right) \right) \\ &\geq LS_f(\boldsymbol{x}) \ . \end{aligned}$$

Next we show that $S^*_{f,\beta}(\boldsymbol{x})$ is β-smooth, i.e,

$$S^*_{f,\beta}(\boldsymbol{y}) \geq e^{-\beta} S^*_{f,\beta}(\boldsymbol{x})$$

for all neighbouring databases $\boldsymbol{x}, \boldsymbol{y}$ (hence, $d(\boldsymbol{x}, \boldsymbol{y}) = 1$). Let $\tilde{\boldsymbol{x}} \in \mathcal{D}$ such that $S^*_{f,\beta}(\boldsymbol{x}) = \Delta^{(\text{local})} f(\tilde{\boldsymbol{x}}) \cdot e^{-\beta d(\boldsymbol{x}, \tilde{\boldsymbol{x}})}$. Such $\tilde{\boldsymbol{x}}$ exists: $\tilde{\boldsymbol{x}} \neq \boldsymbol{x}$ whenever $\beta > 0$. Using the triangle inequality, we know that $d(\boldsymbol{y}, \tilde{\boldsymbol{x}}) \leq d(\boldsymbol{x}, \tilde{\boldsymbol{x}}) + 1$, therefore,

$$\begin{aligned} S^*_{f,\beta}(\boldsymbol{y}) &\geq S^*_{f,\beta}(\tilde{\boldsymbol{x}}) \cdot e^{-\beta d(\boldsymbol{y}, \tilde{\boldsymbol{x}})} \\ &\geq S^*_{f,\beta}(\tilde{\boldsymbol{x}}) \cdot e^{-\beta (d(\boldsymbol{x}, \tilde{\boldsymbol{x}}) - 1)} \\ &= e^{\beta} \cdot S^*_{f,\beta}(\boldsymbol{x}) \ . \end{aligned}$$

Let S be a function that satisfies Definition 5.2.2. To conclude the proof, it is necessary to demonstrate that $S(\boldsymbol{x}) \geq S^*_{f,\beta}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{D}$. We do this by showing that $S(\boldsymbol{x}) \geq \Delta^{(\text{local})} f(\boldsymbol{y}) \cdot e^{-\beta d(\boldsymbol{x}, \boldsymbol{y})}$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$, where $d(\boldsymbol{x}, \boldsymbol{y})$ is the distance between $\boldsymbol{x}, \boldsymbol{y}$. This is done by induction on the value of $d(\boldsymbol{x}, \boldsymbol{y})$.

The base case, $S(\boldsymbol{x}) \geq \Delta^{(\text{local})} f(\boldsymbol{x})$ is the first requirement in Definition 5.2.2. For the induction step, suppose that $S(\tilde{\boldsymbol{x}}) \geq \Delta^{(\text{local})} f(\boldsymbol{y}) \cdot e^{-\beta d(\tilde{\boldsymbol{x}}, \boldsymbol{y})}$ for all $d(\tilde{\boldsymbol{x}}, \boldsymbol{y}) = k$. Consider $\boldsymbol{x}, \boldsymbol{y}$ at distance $k+1$. There exists $\tilde{\boldsymbol{x}} : d(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = 1, d(\tilde{\boldsymbol{x}}, \boldsymbol{y}) = k$. Using the second requirement in Definition 5.2.2, we have that $S(\boldsymbol{x}) \geq S(\tilde{\boldsymbol{x}}) \cdot e^{-\beta}$. By applying the induction hypothesis, we obtain the following inequality: $S(\tilde{\boldsymbol{x}}) \geq \Delta^{(\text{local})} f(\boldsymbol{y}) \cdot e^{-\beta d(\tilde{\boldsymbol{x}}, \boldsymbol{y})}$. This establishes that the desired result is indeed true, namely that

$$S(\boldsymbol{x}) \geq \Delta^{(\text{local})} f(\boldsymbol{y}) \cdot e^{-\beta (d(\tilde{\boldsymbol{x}}, \boldsymbol{y}) + 1))} = \Delta^{(\text{local})} f(\boldsymbol{y}) \cdot e^{-\beta d(\boldsymbol{x}, \boldsymbol{y})} \ .$$

□

In what follows, we will demonstrate how the smooth sensitivity concept can be used to achieve approximate differential privacy. To achieve this, it is first necessary to introduce several key notions.

Definition 5.2.5. *Let $\varepsilon, \delta' > 0$. A probability distribution on $\mathbb{R}^d$ with a density h is (α, β)-admissible with respect to a norm $\|\cdot\|$ if for $\alpha = \alpha(\varepsilon, \delta), \beta = \beta(\varepsilon, \delta')$, the following two conditions hold for all $q \in \mathbb{R}^d$ and $\lambda \in \mathbb{R}$ satisfying $||q|| \leq \alpha$ and $|\lambda| \leq \beta$, and for all measurable subsets $B \subseteq \mathbb{R}^d$:*

$$\mathbb{P}(Z \in B) \leq e^{\varepsilon/2} \cdot \mathbb{P}\left(Z \in B + q\right) + \frac{\delta'}{2} \ , \tag{5.3}$$

$$\mathbb{P}\left(Z \in B\right) \leq e^{\varepsilon/2} \cdot \mathbb{P}\left(Z \in e^{\lambda} B\right) + \frac{\delta'}{2} \ , \tag{5.4}$$

where Z is a random variable with the density h.

We will refer to equations (5.3) and (5.4) as, respectively, the *sliding* and the *dilation* property of the density function h.

The definition is well-suited for the proof of approximate differential privacy to work. As expected, the Laplace distribution is the most important example.

Example 5.2.6. For $\varepsilon, \delta' \in (0,1)$, the 1-dimensional Laplace distribution with the density $h(z) = \frac{1}{2}e^{-|z|}$ is a (α, β)-admissible with

$$\alpha = \frac{\varepsilon}{2} \quad \textit{and} \quad \beta = \frac{\varepsilon}{2\ln(2/\delta')} \ .$$

Furthermore, $\delta' = 0$ in (5.3). Thus, in particular, for the standard Laplace we obtain

$$\mathbb{P}(Z \in B) \leq e^{\varepsilon} \cdot \mathbb{P}\left(Z \in B + q\right) \ , \tag{5.5}$$

whenever $|q| \leq \varepsilon$.

The next results demonstrate that the smooth sensitivity may be employed in lieu of the global sensitivity, thereby preserving differential privacy to some extent. It should be noted that there is a cost to this approach. Even when Laplace noise is employed, only approximate differential privacy is achieved, rather than the more stringent pure ε-differential privacy. Furthermore, the parameter space is constrained. For further details, see Remark 5.2.8. This result should be compared to Theorem 4.3.1, which is the Laplace mechanism.

Proposition 5.2.7. *Let $\varepsilon > 0$ and $\delta' \in (0,1)$. Let $f : \mathcal{D} \to \mathbb{R}_+$ and let $S : \mathcal{D} \to \mathbb{R}_+$ be a β-smooth upper bound on the local sensitivity of f. If $\beta \leq \frac{\varepsilon}{2\ln(2/\delta')}$ and $\delta' \in (0,1)$, the randomized output perturbation mechanism*

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z$$

with the Laplace$(0, 2\frac{S(\boldsymbol{x})}{\varepsilon})$-distributed noise Z. Then, for $\boldsymbol{x} \sim \boldsymbol{y}$ and all Borel sets B,

$$\mathbb{P}(O_f(\boldsymbol{x}, Z) \in B) \leq e^{\varepsilon}\mathbb{P}(O_f(\boldsymbol{y}, Z) \in B) + \frac{\delta'}{2}(e^{\varepsilon/2} + 1) \ .$$

Proof. Let the database $\boldsymbol{x}$ be fixed. Assume that Z has the density h. To shorten the notation, let

$$\mathcal{A}(\boldsymbol{x}) := f(\boldsymbol{x}) + Z = f(\boldsymbol{x}) + \frac{2S(\boldsymbol{x})}{\varepsilon} \cdot Z' \ ,$$

where $Z' \sim$ Laplace(1). We need to show that for all $\boldsymbol{y} \sim \boldsymbol{x} \in \mathcal{D}$, and all Borel sets B in $\mathbb{R}$,

$$\mathbb{P}(\mathcal{A}(\boldsymbol{x}) \in B) \leq e^{\varepsilon}\mathbb{P}(\mathcal{A}(\boldsymbol{y}) \in B) + \frac{\delta'}{2}(e^{\varepsilon/2} + 1) \ .$$

For a Borel set $B \subset \mathbb{R}$ and real numbers a, b, we define $aB + b = \{ax + b : x \in B\}$. Let us denote $N(\boldsymbol{x}) = \frac{2S(\boldsymbol{x})}{\varepsilon}$, and observe that $\mathcal{A}(\boldsymbol{x}) \in B \iff Z' \in B_1$, where $B_1 = \frac{B - f(\boldsymbol{x})}{N(\boldsymbol{x})}$. Let $B_2 = B_1 + \frac{f(\boldsymbol{x}) - f(\boldsymbol{y})}{N(\boldsymbol{x})} = \frac{B - f(\boldsymbol{y})}{N(\boldsymbol{x})}$, and let $B_3 = B_2 \cdot \frac{N(\boldsymbol{x})}{N(\boldsymbol{y})} = \frac{B - f(\boldsymbol{y})}{N(\boldsymbol{y})}$. Then we can write

$$\begin{aligned}
\mathbb{P}(\mathcal{A}(\boldsymbol{x}) \in B) &= \mathbb{P}(Z' \in B_1) \\
&\leq \mathbb{P}\left(Z' \in B_1 + \frac{f(\boldsymbol{x}) - f(\boldsymbol{y})}{N(\boldsymbol{x})}\right) \cdot e^{\varepsilon/2} + \frac{\delta'}{2} \\
&= \mathbb{P}(Z' \in B_2) \cdot e^{\varepsilon/2} + \frac{\delta'}{2} \\
&\leq \mathbb{P}(Z' \in B_3) \cdot e^{\varepsilon} + \frac{\delta'}{2} \cdot e^{\varepsilon/2} + \frac{\delta'}{2} \,.
\end{aligned}$$

The first inequality holds since h satisfies the sliding property (5.3) and since

$$\frac{\|f(\boldsymbol{y}) - f(\boldsymbol{x})\|}{N(\boldsymbol{x})} = \frac{\varepsilon}{2} \cdot \frac{\|f(\boldsymbol{y}) - f(\boldsymbol{x})\|}{S(\boldsymbol{x})} \leq \frac{\varepsilon}{2} \cdot \frac{\|f(\boldsymbol{y}) - f(\boldsymbol{x})\|}{LS_f(\boldsymbol{x})} \leq \frac{\varepsilon}{2} \,. \tag{5.6}$$

Thus, the sliding property is used with $\alpha = \varepsilon/2$. The second inequality holds since h satisfies the dilation property (5.4). Also, S is $\beta-$smooth, which implies that $\left|\ln \frac{N(\boldsymbol{x})}{N(\boldsymbol{y})}\right| = \left|\ln \frac{S(\boldsymbol{x})}{S(\boldsymbol{y})}\right| \leq \left|\ln e^{\pm\beta}\right| \leq \beta$. □

Remark 5.2.8. It should be noted that in the original paper [39], the authors obtained the bound $\mathbb{P}(\mathcal{A}(\boldsymbol{y}) \in B) \cdot e^{\varepsilon} + \delta$, and concluded that approximate differential privacy could be achieved. However, this statement is clearly incorrect, and the corrected statement appears below.

Corollary 5.2.9. *Let $\delta \in [0, 1]$. Under the conditions of Proposition 5.2.7 the algorithm is (ε, δ)-differentially private whenever*

$$\frac{\delta'}{2}(e^{\varepsilon/2} + 1) \leq \delta < 1 \,.$$

Remark 5.2.10. It can be demonstrated that, in fact, $\delta < 1$, which in turn yields constraints on ε. To illustrate this, consider the case where $\delta = 0.01$. In this instance, it follows that $\varepsilon < 2\ln(200)$. This further illustrated in the following section.

Remark 5.2.11. In real-world applications, the value δ must be sufficiently small. This implies that β must also be small. This leads to the conclusion that the induced smooth sensitivity will be close to the smooth sensitivity. From this perspective, the use of the global sensitivity can be considered a minimal improvement. What is important here is that the smooth sensitivity is computable for the given database and the given query.

Proposition 5.2.7 and Corollary 5.2.9 indicate that the addition of Laplace noise with smooth sensitivity yields approximate differential privacy instead of ε-differential privacy. Two questions arise here.

- *Do distributions exist that achieve ε-differential privacy with the smooth sensitivity?*
- *Can Normal noise with the smooth sensitivity be used?*

Both questions can be answered affirmatively. The following observations are presented without a formal proof. Let S be a β-smooth sensitivity.

- Let $\varepsilon > 0$. Let $\beta \leq \frac{\varepsilon}{2(\gamma+1)}$ and $\gamma > 1$. Assume that Z is sampled from the density $h(z) \propto \frac{1}{1+|z|^\gamma}$. Let $c_\varepsilon(\boldsymbol{x}) = \frac{2(\gamma+1)S(\boldsymbol{x})}{\varepsilon}$. The randomized output perturbation mechanism
$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + c_\varepsilon(\boldsymbol{x})Z$$
is ε- differentially private.
- Let $\varepsilon > 0$ and $\delta \in (0,1)$. Take $\beta = \frac{\varepsilon}{2\ln(2/\delta)}$. Let $d_\varepsilon(\boldsymbol{x}) = 5\sqrt{2\ln(2/\delta)}\frac{S(\boldsymbol{x})}{\varepsilon}$. Assume that Z is standard normal. The randomized output perturbation mechanism
$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + d_\varepsilon(\boldsymbol{x})Z$$
is (ε, δ)-differentially private, provided the restrictions from Remark 5.2.10 are fulfilled.

The last item should be compared to Theorem 4.4.3, where we add the noise of the form $c_\varepsilon Z$, where $c_\varepsilon > \sqrt{2\ln(1.25/\delta)}\Delta f/\varepsilon$. Since the smooth sensitivity will be close to the global sensitivity, Δf, we observe that for small δ, it is necessary to add more noise when using the smooth sensitivity than when using the global sensitivity.

Computation of smooth sensitivity. As defined in Definition 5.2.3, the smooth sensitivity is not computable, since it uses all of the possible databases $\boldsymbol{y}$. For some specific queries f, the smooth sensitivity can be computed approximately.

Definition 5.2.12. *Let $\boldsymbol{x}$ be a database. The local sensitivity of f at **distance k** is*
$$A_f^{(k)}(\boldsymbol{x}) = \max_{\boldsymbol{y}\in\mathcal{D}:d(\boldsymbol{x},\boldsymbol{y})\leq k} \Delta^{(\text{local})} f(\boldsymbol{y}) \, .$$

We would like to point out that in order to compute $A_f^{(k)}(\boldsymbol{x})$ we still need to know all the possible databases $\boldsymbol{y}$. Note that
$$A_f^{(0)}(\boldsymbol{x}) = \Delta^{(\text{local})} f(\boldsymbol{x}) \, .$$

Now the smooth sensitivity can be approximated in terms of $A_f^{(k)}$:

$$\begin{aligned}\widetilde{S}^*_{f,\beta}(\boldsymbol{x}) &= \max_{k=0,1,\dots,n} e^{-k\beta}\left(\max_{\boldsymbol{y}:d(\boldsymbol{x},\boldsymbol{y})=k}\Delta^{(\text{local})}f(\boldsymbol{y})\right)\\ &= \max_{k=0,1,\dots,n} e^{-k\beta}A_f^{(k)}(\boldsymbol{x}) = \max\left(\Delta^{(\text{local})}f(\boldsymbol{x}), \max_{k=1,\dots,n} e^{-k\beta}A_f^{(k)}(\boldsymbol{x})\right)\ .\end{aligned}$$

We note that the latter equation gives $\widetilde{S}^*_{f,\beta}(\boldsymbol{x}) \geq \Delta^{(\text{local})}f(\boldsymbol{x})$. Hence, looking at the proof of Proposition 5.2.7 we notice that $\widetilde{S}^*_{f,\beta}(\boldsymbol{x})$ yields approximate differential privacy as well.

Example 5.2.13. Let $f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n x_i$. Recall from Example 2.1.3 that the local sensitivity is $\frac{1}{n}\max|x_i|$. When $d(\boldsymbol{x},\boldsymbol{y}) = 1$, we have

$$A^{(1)}(\boldsymbol{x}) = \max\left|\frac{1}{n}\sum_{i=1}^n x_i - \frac{1}{n}\sum_{i=1,i\neq j}^n x_i - \frac{1}{n}y_j\right| = \frac{1}{n}\max|x_j - y_j|\ ,$$

where the maximum is understood to be taking the maximum over all rows j in the database and all possible entries y_j coming from the original population. Thus, if the original population is constrained to $[0,\Lambda]$, then $A^{(1)}(\boldsymbol{x}) = \max_j \max\{x_j, |x_j - \Lambda|\}/n$. When $d(\boldsymbol{x},\boldsymbol{y}) = k$, we have

$$A^{(k)}(\boldsymbol{x}) = \max\left|\frac{1}{n}\sum_{i=1}^n x_i - \frac{1}{n}\sum_{i=1,i\neq J_k}^n x_i - \frac{1}{n}\sum_{i\in J_k} y_i\right|\ ,$$

where the max is taken again over all k-tuples $J_k = \{(j_1,\dots,j_k) \in \{1,\dots,n\}^k\}$ and all the entries y_j. In particular, the smooth sensitivity can be calculated when sampling from a bounded population. However, it is not possible to compute the smooth sensitivity in an unbounded case. Therefore, the smooth sensitivity is not a particularly useful measure in the context of the mean query.

The next example illustrates the utility of the smooth sensitivity in addressing median queries.

Example 5.2.14. In Example 2.1.3, we demonstrated that for the function $f(\boldsymbol{x}) = \text{median}(x_1,\dots,x_n)$ (assuming that n is odd and $m = \frac{n+1}{2}$), the local and global sensitivity are, respectively:

$$\Delta^{(\text{local})}f(\boldsymbol{x}) = \max(x_{m+1} - x_m, x_m - x_{m-1})\ ,$$
$$\Delta f = \Lambda_2 + \Lambda_1\ .$$

Then

$$\widetilde{S}^*_{f,\beta}(\boldsymbol{x}) = \max_{k=0,\dots,n}\left(e^{-k\beta}\cdot\max_{t=0,\dots,k+1}(x_{m+t} - x_{m+t-k-1})\right)\ . \tag{5.7}$$

It is worth noting that the smooth sensitivity of the median can be computed based on the data in time $O(n^2)$.

Example 5.2.15. Consider again the following example:

- Dataset 1: $x_1 = \cdots = x_m = 0$ and $x_{m+1} = \cdots = x_n = \Lambda$;
- Dataset 2: $x_1 = \cdots = x_{m+1} = 0$ and $x_{m+2} = \cdots = x_n = \Lambda$.

In light of Example 5.2.1, we will check if the smooth sensitivity is zero for one dataset and non-zero for another data.

Let $a_{t,k} = x_{m+t} - x_{m+t-k-1}$. We calculate $a_{t,k}$ for values of t and k for Dataset 1 and Dataset 2.
When $t = 0$:

$$a_{0,k} = x_{m+1} - x_{m-k-1} = 0 \quad \forall k \text{ for both datasets.}$$

When $t = 1$:

$$a_{1,k} = x_{m+1} - x_{m-k}$$

We calculate this expression for different values of k.
For $k = 0$,

$$\begin{aligned} a_{1,0} = \Lambda - 0 = \Lambda \quad & \text{for Dataset 1,} \\ = 0 \quad & \text{for Dataset 2.} \end{aligned}$$

For $k = 1$:

$$\begin{aligned} a_{1,1} = \Lambda \quad & \text{for Dataset 1,} \\ = 0 \quad & \text{for Dataset 2.} \end{aligned}$$

For the case where $a_{2,k} = x_{m+2} - x_{m-k-1}$, we observe that for Dataset 1, $a_{2,0} = 0$ and $a_{2,1} = \Lambda$, while for Dataset 2, $a_{2,0} = \Lambda$ and $a_{2,1} = 0$.

Therefore, the values within the brackets of equation (5.7) alternate between 0 and Λ, and thus the issue previously encountered in Example 5.2.1 is no longer present.

Summary for smooth sensitivity. In conclusion, the smooth sensitivity is a concept in differential privacy that provides a more nuanced approach to measuring the sensitivity of a function to changes in its input data. This approach is particularly useful in situations where the global sensitivity might be overly conservative (or not possible to calculate). In meaningful applications, a smooth sensitivity is nearly equivalent to the global sensitivity. Conversely, a smooth sensitivity can be calculated based on the database, whereas the global sensitivity cannot be. Nevertheless, there is a trade-off associated with the use of the smooth sensitivity. Instead of achieving ε-differential privacy, only approximate differential privacy can be achieved.

5.2.2 Towards general sensitivity

Recall that the global and local sensitivities are related by

$$\Delta f = \max(\Delta^{(\text{local})} f(\boldsymbol{x}), \max_{\boldsymbol{y} \neq \boldsymbol{x}} \Delta^{(\text{local})} f(\boldsymbol{y})) \ .$$

We consider the mean query. The global sensitivity may be infinite if the database comes from an unbounded population. Even if the population is bounded (say, between 0 and Λ), the global sensitivity is Λ, and the resulting variance of the Laplace mechanism is proportional to Λ^2, which can be huge. This has a negative effect on data utility. On the other hand, a much smaller local sensitivity violates differential privacy because it can leak information about the database.

The idea is to find a (possibly random) "sensitivity" that lies between the local and the global sensitivities, preserves some properties of the database, and most importantly, is computable. This concept will be informally referred to as the *general sensitivity*. One such example is the smooth sensitivity discussed in the previous section. As indicated there, the approach based on smooth sensitivity still does not solve the problem in case of the mean query.

We are going to focus on queries of the form

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + N_\xi(\boldsymbol{x}) Z' \ ,$$

where $Z' \sim \text{Laplace}(1)$ and $N_\xi(\boldsymbol{x})$ is a "general sensitivity function" that possibly depends on the database $\boldsymbol{x}$ and an auxiliary random variable ξ, independent from the database. It will also depend on the privacy budget ε.

Definition 5.2.16. *Let ξ be a nonnegative random variable, independent of the database. Let $f : \mathcal{D} \to \mathbb{R}_+$ be a query. Set $\varepsilon, \beta > 0$. Consider a (random) function $N_\xi : \mathcal{D} \to \mathbb{R}_+$ such that*

1. $N_\xi(\boldsymbol{x}) > \dfrac{\Delta^{(\text{local})} f(\boldsymbol{x})}{\varepsilon/2}$,
2. *for all $\boldsymbol{y} \sim \boldsymbol{z}$,* $\left| \ln \dfrac{N_\xi(\boldsymbol{y})}{N_\xi(\boldsymbol{z})} \right| \leq \beta < \infty$.

We call N_ξ a general (ε, β, f)-sensitivity.

Let $g_A(q)$ be the density of a random variable A at point q. We want to show that

$$\mathbb{P}(f(\boldsymbol{x}) + N_\xi(\boldsymbol{x}) Z' \in B) \leq e^\varepsilon \mathbb{P}(f(\boldsymbol{y}) + N_\xi(\boldsymbol{y}) Z' \in B) \ .$$

Step 1. Note that the noise, $N_\xi(\boldsymbol{x})Z'$ may not Laplace distributed because of the mixture factor $N_\xi(\boldsymbol{x})$. However, the noise component has the Laplace distribution, conditionally on ξ. Thus, denote by $\mathbb{E}_\xi[\cdot]$ the conditional expectation given ξ. The left hand side of the expression above is

$$\begin{aligned}\mathbb{E}_\xi[\mathbb{P}f(\boldsymbol{x}) + N_\xi(\boldsymbol{x})Z' \in B)] &= \mathbb{E}_\xi\left[\mathbb{P}\left(Z' \in \frac{B - f(\boldsymbol{x})}{N_\xi(\boldsymbol{x})}\right)\right] \\ &= \mathbb{E}_\xi\left[\frac{1}{2}\int_B e^{-\frac{|q-f(\boldsymbol{x})|}{N_\xi(\boldsymbol{x})}}\,dq\right] .\end{aligned}$$

We would like to obtain a similar expression that involves both $f(\boldsymbol{y})$ and $N_\xi(\boldsymbol{y})$. This will be done in two phases.

Step 2. Mimic the proof of differential privacy(see the proof of Theorem 4.3.1.)

First we will replace $f(\boldsymbol{x})$ with $f(\boldsymbol{y})$. We have

$$\begin{aligned}\mathbb{E}_\xi[\mathbb{P}(f(\boldsymbol{x}) + N_\xi(\boldsymbol{x})Z' \in B)] &= \mathbb{E}_\xi\left[\frac{1}{2}\int_B e^{-\frac{|q-f(\boldsymbol{x})|}{N_\xi(\boldsymbol{x})}}\,dq\right] \\ = \mathbb{E}_\xi\left[\frac{1}{2}\int_B \underbrace{e^{\frac{-|q-f(\boldsymbol{x})|+|q-f(\boldsymbol{y})|}{N_\xi(\boldsymbol{x})}}}_{=:I} e^{-\frac{|q-f(\boldsymbol{y})|}{N_\xi(\boldsymbol{x})}}\,dq\right] .\end{aligned}$$

As in the proof of differential privacy, the absolute value of the first part is bounded by

$$|\ln I| \le \left|\frac{|q-f(\boldsymbol{x})| - |q-f(\boldsymbol{y})|}{N_\xi(\boldsymbol{x})}\right| \le \frac{|f(\boldsymbol{x}) - f(\boldsymbol{y})|}{N_\xi(\boldsymbol{x})} \le \frac{\Delta^{(\text{local})}f(\boldsymbol{x})}{N_\xi(\boldsymbol{x})}.$$

Formally, the term in question depends on the database $\boldsymbol{x}$ and neighbouring databases. The goal is to establish a bound on $|I|$ by a constant that is independent of the databases and depends on the value of ε. Recall now that $\boldsymbol{x}$ is fixed. At the same time, the numerator of the last expression is bounded by

$$\sup_{\boldsymbol{y}:\boldsymbol{y}\sim\boldsymbol{x}} |f(\boldsymbol{x}) - f(\boldsymbol{y})| = \Delta^{(\text{local})}f(\boldsymbol{x}) .$$

Thus, whenever

$$N_\xi(\boldsymbol{x}) > \frac{\Delta^{(\text{local})}f(\boldsymbol{x})}{\varepsilon/2} , \tag{5.8}$$

we get:

$$\mathbb{E}_\xi[\mathbb{P}(f(\boldsymbol{x}) + N_\xi(\boldsymbol{x})Z' \in B)] \le e^{\varepsilon/2}\frac{1}{2}\mathbb{E}_\xi\left[\int_B e^{-\frac{|q-f(\boldsymbol{y})|}{N_\xi(\boldsymbol{x})}}\,dq\right] .$$

Step 3. The problem is that the right-hand side of the equation is dependent on $\boldsymbol{x}$ through $N_\xi(\boldsymbol{x})$. In order to resolve this, we need to replace it with $N_\xi(\boldsymbol{y})$. The term can be rewritten as follows:

$$\begin{aligned}&\frac{1}{2}\mathbb{E}_\xi\left[\int_B e^{-\frac{|q-f(\boldsymbol{y})|}{N_\xi(\boldsymbol{x})}}\,dq\right] = \frac{1}{2}\mathbb{E}_\xi\left[\int_B e^{-\frac{|q-f(\boldsymbol{y})|}{N_\xi(\boldsymbol{y})}\cdot\frac{N_\xi(\boldsymbol{y})}{N_\xi(\boldsymbol{x})}}\,dq\right]\\ &= \mathbb{E}_\xi\left[\mathbb{P}\left(Z' \in \frac{B-f(\boldsymbol{y})}{N_\xi(\boldsymbol{x})}\frac{N_\xi(\boldsymbol{y})}{N_\xi(\boldsymbol{x})}\right)\right].\end{aligned}$$

Now, assume there exists $\beta < \infty$ such that for all databases $\boldsymbol{y} \sim \boldsymbol{z}$ (related to $\boldsymbol{x}$ or not)

$$\left|\ln\frac{N_\xi(\boldsymbol{y})}{N_\xi(\boldsymbol{z})}\right| \le \beta\,. \tag{5.9}$$

It can be reasonably assumed that a small value of β will result in a reduced level of privacy leakage. A large value of β results in a significant degree of privacy leakage. The above bound prevents N_ξ from being too small. Formally setting $\beta = \infty$ ($N_\xi(\boldsymbol{y}) = 0$) results in a breakdown of differential privacy, analogous to the scenario depicted in the example of the median query with local sensitivity, where it is possible to obtain zero.

When (5.9) holds, due to the dilation property (5.4) of the Laplace distribution,

$$\begin{aligned}&\frac{1}{2}\mathbb{E}_\xi\left[\int_B e^{-\frac{|q-f(\boldsymbol{y})|}{N_\xi(\boldsymbol{x})}}\,dq\right] = \mathbb{E}_\xi\left[\mathbb{P}\left(Z' \in \frac{B-f(\boldsymbol{y})}{N_\xi(\boldsymbol{x})}\frac{N_\xi(\boldsymbol{y})}{N_\xi(\boldsymbol{x})}\right)\right]\\ &\le e^{\epsilon/2}\mathbb{E}_\xi\left[\mathbb{P}\left(Z' \in \frac{B-f(\boldsymbol{y})}{N_\xi(\boldsymbol{x})}\right)\right] + \delta'/2\,.\end{aligned}$$

Summarizing, we obtain the following result.

Theorem 5.2.17. *Let $\varepsilon, \beta > 0$. Let $f : \mathcal{D} \to \mathbb{R}_+$. Let N_ξ be a general (ε, β, f)-sensitivity. Consider the randomized output perturbation mechanism*

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + N_\xi(\boldsymbol{x})Z'$$

with the Laplace(1)-distributed noise Z'. Then there exists $\delta' > 0$ such that for $\boldsymbol{x} \sim \boldsymbol{y}$ and all Borel sets B,

$$\mathbb{P}(O_f(\boldsymbol{x}, Z) \in B) \le e^{\varepsilon}\mathbb{P}(O_f(\boldsymbol{y}, Z) \in B) + e^{\varepsilon/2}\delta'/2\,.$$

Example 5.2.18. Consider for example:

- $N_\xi(\boldsymbol{x}) = 2\Delta f/\varepsilon$. Then we reduce the problem to the one of classical differential privacy.

- $N_\xi(\boldsymbol{x}) = 2S(\boldsymbol{x})/\varepsilon$, where $S(\boldsymbol{x})$ is the smooth sensitivity.

We note that the local sensitivity cannot be taken as $N_\xi(\boldsymbol{x})$, as the condition in (5.9) is violated. See Example 5.2.1.

Two questions arise from the above calculations:

1. Set $\delta = e^{\varepsilon/2}\delta'/2$. *What is the precise definition of the privacy leakage denoted by* δ*?*

2. *What do we gain from a data utility point of view?*

In response to the first question, we may utilize Example 5.2.6 to conclude that the following relationship holds

$$\delta = \delta(\varepsilon, \beta) = e^{\varepsilon/2} e^{-\varepsilon/(2\beta)} \,. \tag{5.10}$$

We note that a very interesting feature is present here. If $\beta > 1$, then δ is an increasing function of ε. This is an entirely intuitive conclusion, given that an increase in the value of ε corresponds to a reduction in the level of privacy. Conversely, if $\beta < 1$, then δ is a decreasing function of ε. Hence, in this region where $\beta < 1$, an increase in ε results in a decrease in the privacy "in the ε part," yet an improvement in the δ part. This phenomenon is counterintuitive, yet it is precisely what is observed in the examples that follows.

In order to respond to the second question, we will now consider the candidates for $N_\xi(\boldsymbol{x})$. To do this, we must check wether we can find an upper bound for the expression:

$$I_\xi(\boldsymbol{z}, \boldsymbol{y}) := \ln \frac{N_\xi(\boldsymbol{y})}{N_\xi(\boldsymbol{z})} \,. \tag{5.11}$$

Candidate 1. Assume that $\Delta f < \infty$ is known. We choose the following candidate:

$$N_\xi(\boldsymbol{x}) = N(\boldsymbol{x}) = \frac{2}{\varepsilon}\left(a\Delta^{(\text{local})} f(\boldsymbol{x}) + b\Delta f\right) \,,$$

where $a + b = 1$, $a \in [0, 1)$, and $b \in (0, 1]$. Thus, we use a *weighted sensitivity*. Then for $I = I_\xi$,

$$\begin{aligned}
\exp(I(\boldsymbol{x}, \boldsymbol{y})) &= \frac{a\Delta^{(\text{local})} f(\boldsymbol{x}) + b\Delta f}{aLS_f(\boldsymbol{y}) + b\Delta f} \mathbb{1}\{\Delta^{(\text{local})} f(\boldsymbol{x}) < \Delta f\} \\
&> \frac{a\Delta^{(\text{local})} f(\boldsymbol{x}) + b\Delta f}{a\Delta f + b\Delta f} \\
&> \frac{b\Delta f}{(a+b)\Delta f} = b \,.
\end{aligned}$$

Thus, $|I(\boldsymbol{x}, \boldsymbol{y})| \leq |\ln b|$. Hence, $\beta = |\ln b|$. As the weight assigned to the global sensitivity,$b \in (0, 1]$ increases and approaches 1, the resulting value of β decreases, leading to a reduction in privacy leakage. This is a logical consequence of the relationship between the two variables. Formally, if $b = 1$, then $\beta = 0$, while $\lim_{\beta \to 0} \delta(\varepsilon, \beta) = 0$ in agreement with the principles of classical differential privacy.

Candidate 2. In the event that the global sensitivity is unknown, the following expression may be employed:

$$N_\xi(\boldsymbol{x}) = \frac{2}{\varepsilon} \left(a\xi + b\Delta^{(\text{local})} f(\boldsymbol{x}) \right) \ ,$$

where ξ is a nonnegative random variable, and the parameters $a + b = 1$, $a \in [0, 1)$, and $b \in (0, 1]$. It can be argued that the random variable ξ serves to "blur" the local sensitivity, given that the latter violates differential privacy. Assume that the global sensitivity, $\Delta f < \infty$, is finite. It should be noted that the local sensitivity is bounded by the global sensitivity. Moreover, the local sensitivity is non-negative. In such a case,

$$\begin{aligned}
\exp(I_\xi(\boldsymbol{x}, \boldsymbol{y})) &= \frac{a\xi + b\Delta^{(\text{local})} f(\boldsymbol{x})}{a\xi + b\Delta^{(\text{local})} f(\boldsymbol{y})} \{\xi < \Delta^{(\text{local})} f(\boldsymbol{x})\} + \frac{a\xi + b\Delta^{(\text{local})} f(\boldsymbol{x})}{a\xi + b\Delta^{(\text{local})} f(\boldsymbol{y})} \{\xi > \Delta^{(\text{local})} f(\boldsymbol{x})\} \\
&> \frac{(a+b)\xi}{a\xi + b\Delta^{(\text{local})} f(\boldsymbol{y})} \{\xi < \Delta^{(\text{local})} f(\boldsymbol{y})\} + \frac{a\xi + b\Delta^{(\text{local})} f(\boldsymbol{x})}{a\xi + b\Delta^{(\text{local})} f(\boldsymbol{y})} \{\xi > \Delta^{(\text{local})} f(\boldsymbol{x})\} \\
&> \frac{\xi}{a\xi + b\Delta f} \{\xi < \Delta^{(\text{local})} f(\boldsymbol{x})\} + \frac{a\xi}{a\xi + b\Delta f} \{\xi > \Delta^{(\text{local})} f(\boldsymbol{x})\} \\
&\geq \frac{a\xi}{a\xi + b\Delta f} \ . \qquad (5.12)
\end{aligned}$$

If the support of the random variable ξ is separated from zero, then the latter bound is also separated from zero. The greater the separation from zero, the bigger the lower bound, and thus the smaller the privacy leakage.

Data utility. We can measure data utility of a (ε, δ)-differentially private random mechanism through the mean squared error. For example if $O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z$ with $Z \sim \text{Laplace}(\Delta f/\varepsilon)$, then the mean squared error is equal to the variance, which in turn is given by

$$\frac{2(\Delta f)^2}{\varepsilon^2}.$$

Data utility for Candidate 1. Now, for the random mechanism

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + N_\xi(\boldsymbol{x}) Z'$$

with $N_\xi(\boldsymbol{x}) = N(\boldsymbol{x}) = \frac{2}{\varepsilon}(a\Delta^{(\text{local})}f(\boldsymbol{x}) + b\Delta f)$ the variance equals

$$\begin{aligned}\operatorname{Var}(N(\boldsymbol{x})Z') &= \frac{4}{\varepsilon^2}(a\Delta^{(\text{local})}f(\boldsymbol{x}) + b\Delta f)^2 \\ &= \frac{4}{\varepsilon^2}b^2(\Delta f)^2 + \frac{4}{\varepsilon^2}a^2(\Delta^{(\text{local})}f(\boldsymbol{x}))^2 + \frac{8ab}{\varepsilon^2}\Delta^{(\text{local})}f(\boldsymbol{x})\Delta f\ . \end{aligned} \tag{5.13}$$

- When $b = 1$, the result is equivalent to the doubled variance for classical differential privacy with the global sensitivity, as previously discussed.
- When $a = 1$, it reduces to the doubled variance for classical differential privacy with the local sensitivity (as discussed in the previous section, differential privacy may fail here).

If $\Delta^{(\text{local})}f(\boldsymbol{x})$ is much smaller then Δf, then by choosing a close to 1, we can greatly improve data utility as compared to classical differential privacy. Of course, as mentioned above, the bigger the value of a, the smaller the value of b, and consequently, the bigger the privacy leakage δ.

Example 5.2.19. In this example we illustrate our weighted sensitivity approach.

We generate a database $\boldsymbol{x}$ of $n = 100$ values from a distribution with a bounded support. The first choice is the uniform on $[0, \Lambda]$, while the second choice is a version of a truncated exponential distribution. The procedure begins with generating a sample from an unbounded exponential. Subsequently, all values exceeding the threshold value of Λ are reduced to Λ.

We use the median query, thereby establishing that the global sensitivity, as defined in Example 2.1.3, is equal to Λ.

For the generated dataset we also calculate the local sensitivity; see Example 2.1.3. The procedure was repeated 5 times, with each iteration resulting gin a distinct local sensitivity. For each case, the data utility was calculated using the formula (5.13) and plotted against values of b. The results are presented in Figure 5.1 and Figure 5.2, which display the weighted sensitivity in the uniform and exponential cases, respectively.

The benchmark is the data utility obtained with classical differential privacy. It is displayed as the dashed horizontal line. The coloured solid curves indicate different repetitions of our experiment. When the curve is below the benchmark, we have an improvement in data utility. The admissible range of b is defined as $(0, b_0^{(\text{data utility})}(\boldsymbol{x})]$. It should be noted that b_0 depends on the generated dataset. For example, for the uniform case we can read from Figure 5.1 that in one of the experiments (depicted by the pink curve), the admissible range of b was $(0, 0.7]$.

Next, we analyze the effect on δ-privacy. We use the formula (5.10) with $\beta = |\ln b|$, leading to

$$\delta = \delta(\varepsilon, b) = e^{\varepsilon/2} e^{-\varepsilon/(2|\ln b|)} .$$

First, we want δ to be below the prescribed threshold, say 0.1. Next, note that δ is a decreasing function of ε whenever $b > \exp(-1)$ (as indicated above, this is somehow counterintuitive). The results are displayed on Figure 5.3. We plotted there δ curves for $\varepsilon \in \{0.5, 1, 2, 10, 15\}$. The resulting δ is admissible whenever it falls below the prescribed threshold. This yields the admissible range of b, denoted by $[b_0^{(\text{privacy})}(\boldsymbol{x}), 1)$.

The admissible range of b from the point of view of both data utility and data privacy is the intersection

$$(0, b_0^{(\text{data ,utility})}(\boldsymbol{x})] \cap [b_0^{(\text{privacy})}(\boldsymbol{x}), 1) .$$

If this intersection is not empty, we can use our weighted sensitivity algorithm. It then keeps the required level of privacy, while improved data utility.

For example, for $\varepsilon = 15$, the admissible range stemming from δ-curves is approximately $[0.45, 1]$ which yields a nonempty intersection with several generated databases $\boldsymbol{x}$.

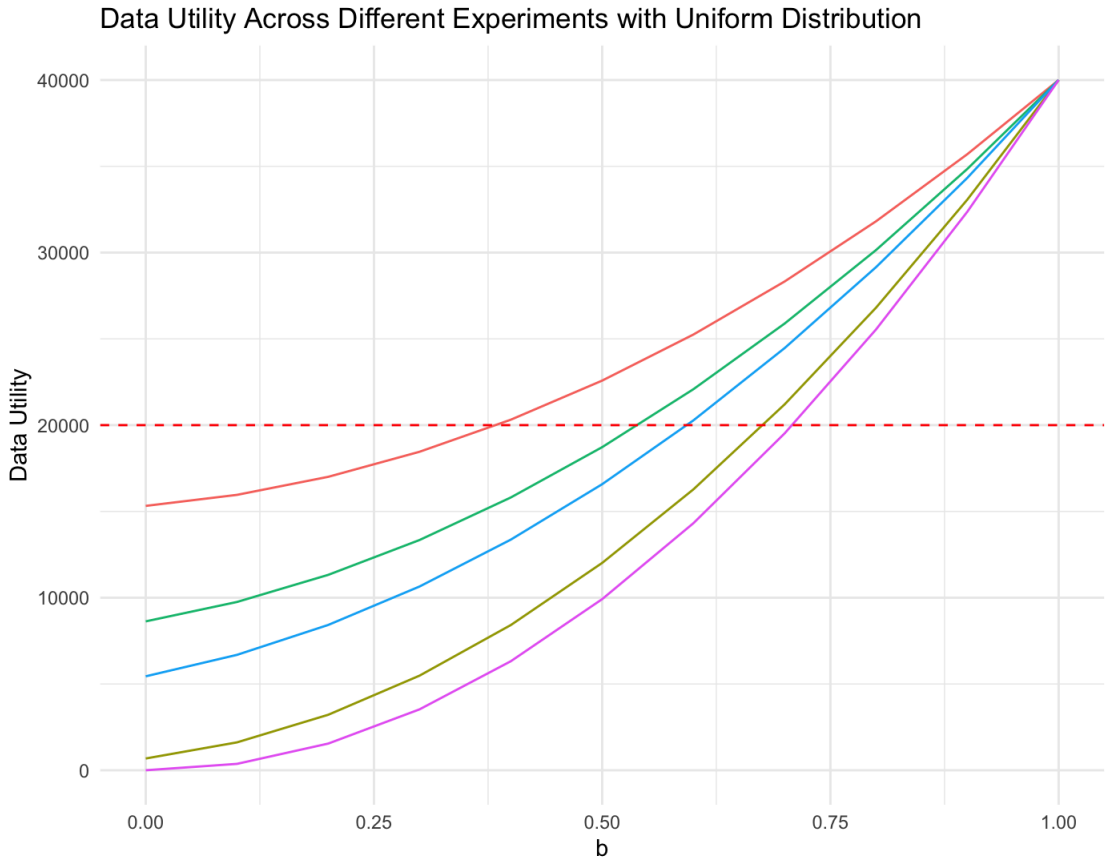

Figure 5.1: Weighted sensitivity - uniform
Uniform case: The illustration demonstrates the utility of data in a weighted sensitivity approach, with each curve corresponding to a distinct simulation of the database, thus yielding disparate weighted sensitivities.

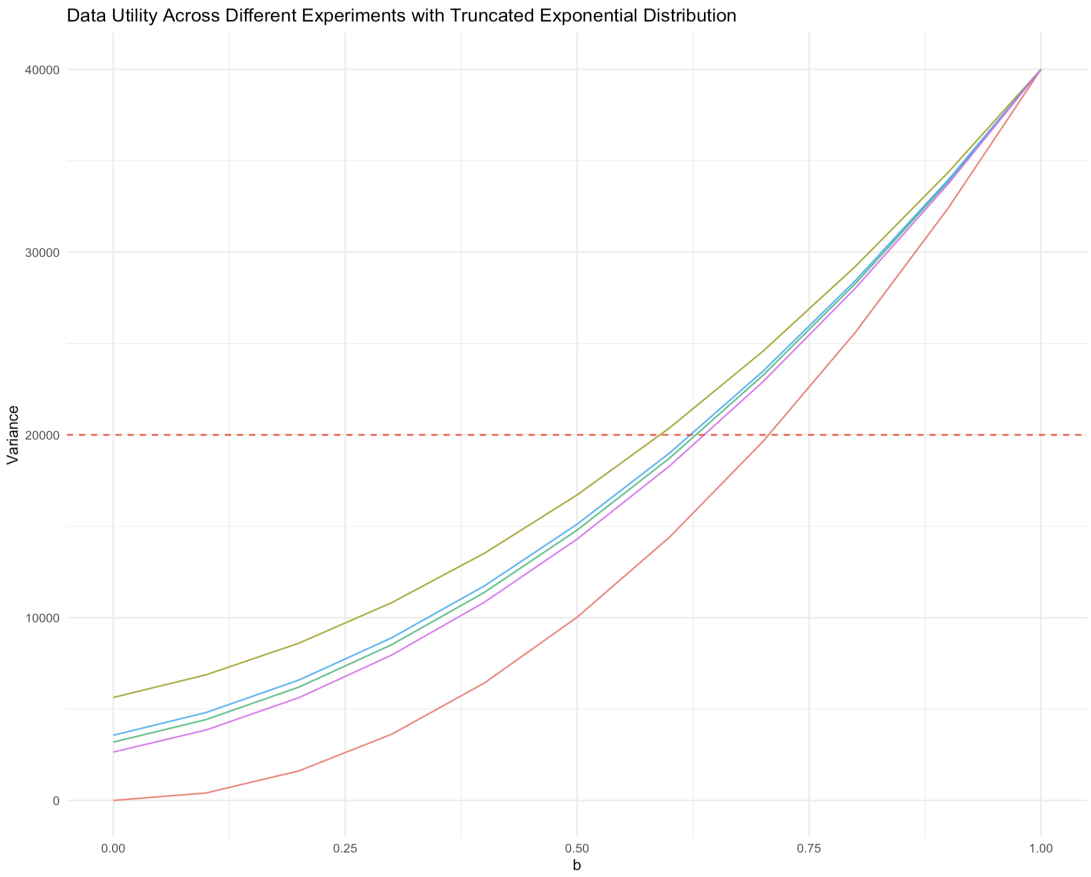

Figure 5.2: Weighted sensitivity - truncated exponential
Exponential case: The illustration demonstrates the utility of data in a weighted sensitivity approach, with each curve corresponding to a distinct simulation of the database, thus yielding disparate weighted sensitivities.

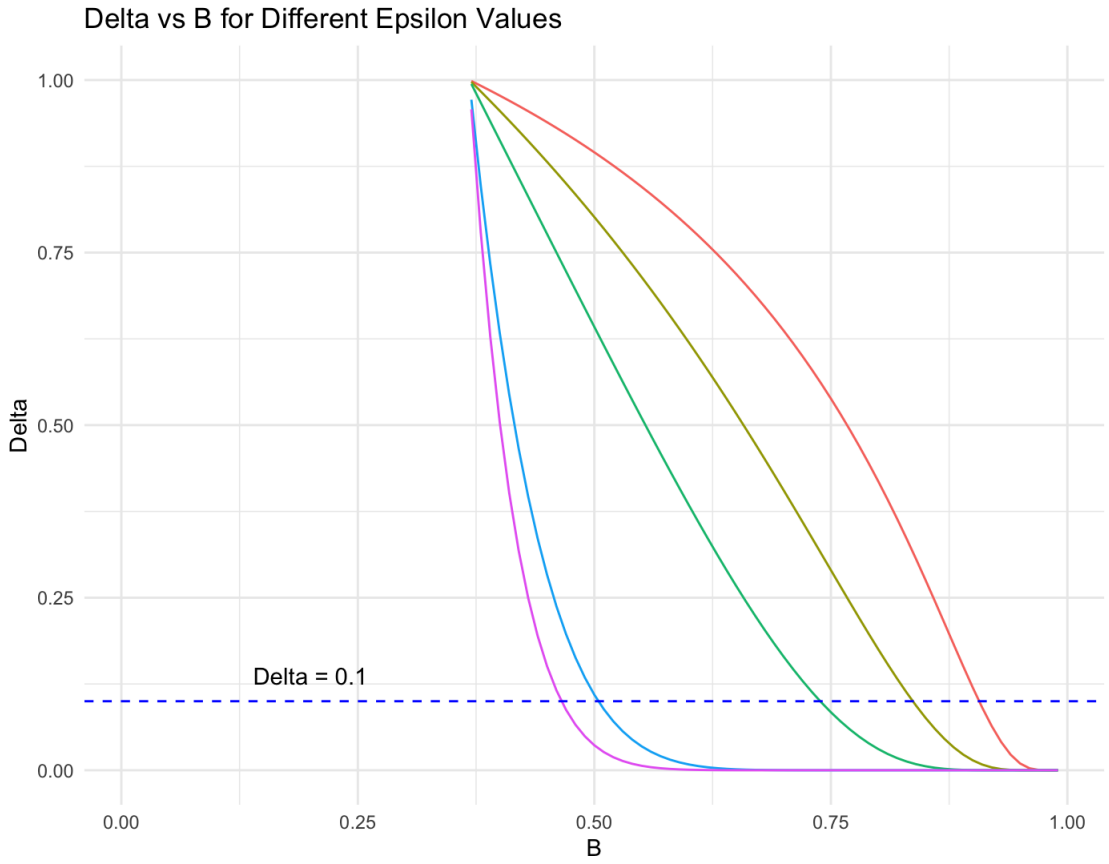

Figure 5.3: δ-privacy
In the weighted sensitivity approach, the concept of δ-privacy is demonstrated through the use of different curves, each corresponding to a specific value of ε. The pink curve corresponds to the value of $\varepsilon = 15$.

Data utility for Candidate 2. For the random mechanism

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + N_\xi(\boldsymbol{x})Z'$$

with

$$N_\xi(\boldsymbol{x}) = N_\xi = \frac{2}{\varepsilon}(a\xi + b\Delta^{(\text{local})}f(\boldsymbol{x}))$$

we have

$$\text{Var}(N_\xi Z') = \text{Var}\left(\frac{2}{\varepsilon}(a\xi + b\Delta^{(\text{local})}f(\boldsymbol{x}))Z'\right) = \frac{4}{\varepsilon^2}\text{Var}\left((a\xi + b\Delta^{(\text{local})}f(\boldsymbol{x}))Z'\right) \ .$$

For any two random variables we have

$$\text{Var}(XY) = \text{Var}(X)\text{Var}(Y) + \text{Var}(X)(\mathbb{E}[Y])^2 + \text{Var}(Y)(\mathbb{E}[X])^2 \ .$$

Here: If $X = a\xi + b\Delta^{(\text{local})} f(\boldsymbol{x})$, $Y = Z'$, $\text{Var}(X) = a^2\text{Var}(\xi)$, $\text{Var}(Z') = 2$, then

$$(\mathbb{E}[a\xi + b\Delta^{(\text{local})} f(\boldsymbol{x})])^2 = (a\mathbb{E}[\xi] + b\Delta^{(\text{local})} f(\boldsymbol{x}))^2 \ ,$$

$$\mathbb{E}[Z'] = 0 \ .$$

Hence, we conclude that

$$\text{Var}(N_\xi(\boldsymbol{x})Z') = \frac{4a^2}{\varepsilon^2}\left\{\text{Var}(\xi) + 2(a\mathbb{E}[\xi] + b\Delta^{(\text{local})} f(\boldsymbol{x}))^2\right\} \ . \tag{5.14}$$

5.3 Mixed Noise Mechanism (MNM)

The classical definition of differential privacy allows for the use of only a Laplace noise. From the statistical inference point of view, it may be desirable to use a normal noise. Gaussian or normal noise is the most well-established in statistical inference, as evidenced by the extensive literature on the topic (see, for example [14]). Deviations from this distributional assumption can require significant adjustments to statistical modelling, for example in the context of maximum likelihood estimation, calculating confidence intervals, hypothesis testing, and so on.

A Gaussian distribution violates differential privacy, yet it aligns with the framework of approximate differential privacy. It is important to recall the results presented in Theorem 4.4.3. The Gaussian output perturbation mechanism fulfills the (ε, δ)-differential privacy whenever the variance $\sigma^2 > 2\Delta f \ln(1.25/\delta)/\varepsilon$. Now, the parameter δ has to be small. This implies that the variance must be substantial, which, in turn, necessitates the addition of a considerable amount of noise.

Meanwhile, for example in biomedical studies, while maintaining participants' privacy, the need to maximize the clinical utility of data is well-established to lessen the burden on trial participants and patients from which data is derived; see [34].

Thus, in this section we propose an approximate differentially private Gaussian mechanism for low sensitivity queries. The contents of this section is based on the author's paper, [8], adapted to the format of the thesis.

We propose a noise mechanism that improves data utility and yields (ε, δ)-differential privacy with the user-controlled small δ. In its simplest possible version, the mechanism adds a specifically chosen normal noise when sensitivity of the query function f is "small" while adding Laplace noise otherwise. As such, we call it a Mixed Noise Mechanism (MNM). It should be emphasized that small values of sensitivity are not uncommon; thus, the need for improved data utility exists. In particular, if f is an estimator of a

population parameter related to the database (for example, the sample mean; the estimator of the population mean), then f is of order $1/n$, where n is the size of the database.

The general idea of the proposed mechanism is to *add normal noise when the data is "conformant" and add Laplace noise when the data is "non-conformant," that is, when significant outliers are present.* The choice between "conformant" and " non-conformant" data is driven by a specific threshold (to be discussed below). In essence, we allow for the use of a Gaussian mechanism with a smaller variance for low-sensitive queries. The new mechanism has very positive impacts on data utility and statistical inference, maintaining the truthfulness of the statistical outputs. We show some useful outcomes of the Mixed Noise Mechanism when studying confidence intervals around low-sensitive queries.

In short, the following statements are provided for the sake of clarity.

- MNM, Laplace, and Gaussian mechanisms achieve a similar level of privacy.
- MNM and Laplace mechanisms perform similarly from the point of view of data utility. At the same time, both the MNM and Laplace mechanism outperform the Gaussian mechanism in our experiments.
- The MNM mechanism has wider applicability from the statistical inference point of view. We illustrate it using confidence intervals.

We note further that this mixed mechanism idea can be used in conjunction with other modifications of a differentially private algorithm.

Data utility perspective. We recall that

$$Z \sim \mathcal{N}(\mu, \sigma^2): \quad \mathbb{E}[(Z-\mu)^p] = \begin{cases} 0 & \text{if } p \text{ is odd}, \\ \sigma^p (p-1)!! & \text{if } p \text{ is even}. \end{cases}$$

$$Z \sim \text{Lap}(b): \quad \mathbb{E}[Z^p] = \begin{cases} 0 & \text{if } p \text{ is odd}, \\ b^p p! & \text{if } p \text{ is even}. \end{cases}$$

Let $Z_1 \sim \mathcal{N}(0, \sigma^2)$ and $Z_2 \sim \text{Lap}(0, b)$. Then we know that $\text{Var}(Z_1) = \sigma^2$ and $\text{Var}(Z_2) = 2b^2$. We choose $\sigma^2 = 2b^2$, so that the Gaussian and Laplace variables have the same variance. Now, $\mathbb{E}[Z_1^4] = 3\sigma^4 = 12b^4$, $\mathbb{E}[Z_2^4] = 24b^4$. Hence, the kurtosis of Laplace is higher than in the case of Normal, implying more variability of Laplace and hence less data utility.

If we decide to use a normal noise, we know that ε-differential privacy is violated, yet (ε, δ)-differential privacy holds. The parameter δ can be interpreted as the probability of information leakage. Consequently, δ should be as small as possible, certainly between 0 and 1. However, if δ is small, then according to Theorem 4.4.3, the variance of the Gaussian noise, called σ^2_{ADP} here, has to be very large (see Figure 5.4). This negatively affects

data utility due to the necessary increase in variability required to ensure differential privacy. In other words, the resulting anonymized query has a large variability.

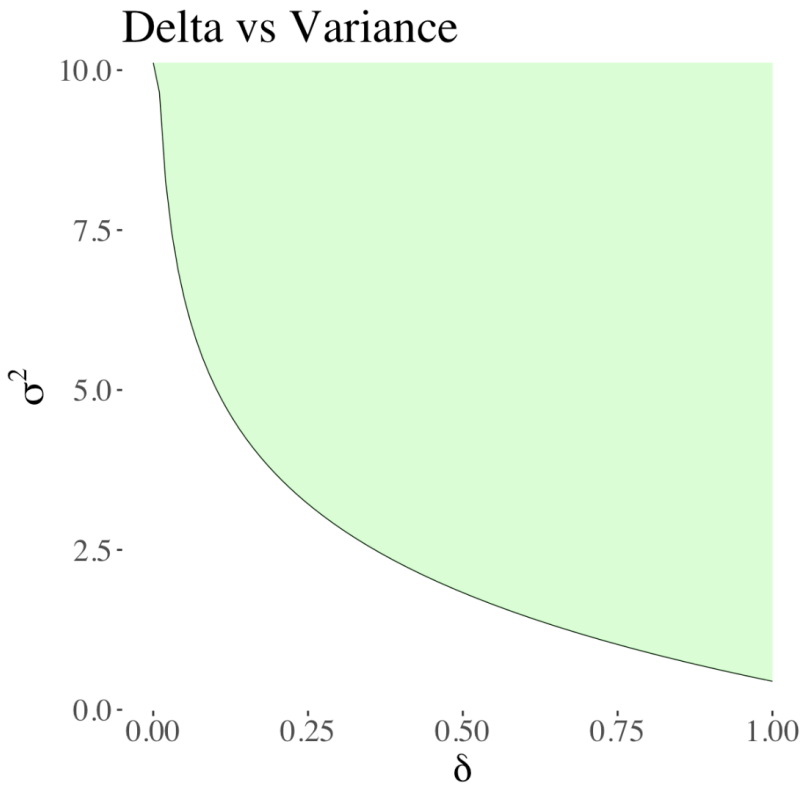

Figure 5.4: δ values for ADP
The green area represents the possible values of σ^2 in Theorem 4.4.3 when $\Delta f = 1, \varepsilon = 1$.

Definition 5.3.1. *Let $f : \mathcal{D} \to \mathbb{R}_+$. Let $t_0 > 0$ and $\sigma_{MNM} > 0$. We call*

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + \mathcal{N}(0, \sigma^2_{MNM})\mathbb{1}\{\Delta f \leq t_0\} + Lap(\Delta f/\varepsilon)\mathbb{1}\{\Delta f > t_0\} \ ,$$

the Mixed Noise Mechanism.

Mixed Noise Mechanism (MNM). In other words, when the sensitivity Δf is large, we choose to add a Laplace noise, while when Δf is small, we choose to add a Gaussian MNM noise with the same variance as in the original formulation of differential privacy. We can ensure that the mechanism is (ε, δ)-differential privacy so that the approach is well justified theoretically. We include a drawing by the author that depicts the idea through Minions (yes, Minions!) in Figure 5.5.

Unlike the approximate differential privacy mechanism, the variance of the Gaussian MNM noise does not depend on δ. We can choose

$$\sigma^2_{MNM} = 2(\Delta f/\varepsilon)^2,$$

so we have the same resulting variance regardless of the noise that is ultimately added. This is helpful when comparing different methods and removing any confusion over what noise is really being added to the dataset. By fixing variance to be equivalent in either method, we can get a true sense of the improvement when implementing our mixed noise approach.

We need to provide a threshold that triggers the mechanism to add either a Laplace or Gaussian noise. The choice of such threshold t_0 is motivated by an inspection of the proof of Theorem A.1 in [17]. We can observe that a Gaussian noise violates ε-differential privacy because of the inability to bound the term

$$\frac{\sigma^2\varepsilon}{\Delta f} - \frac{\Delta f}{2},$$

where σ^2 is the variance of the Gaussian noise added in the (ε, δ)-differential privacy framework. If Δf is large, then $\mathbb{P}(|\mathcal{N}(0, \sigma^2)| > t_0)$ is still large, yielding unacceptable values of δ in the definition of (ε, δ)-differential privacy. On the other hand, if Δf is small, then the probability is small, yielding the required user-chosen values of δ.

Small values of Δf are relatively common, and thus, this violation does not occur often in practice. This can represent a significant improvement in data utility by decreasing the amount of noise needed to achieve (ε, δ)-differential privacy, and using a Gaussian distribution so that methods of statistical inference are preserved and can be utilized in application. We refer to Example 2.1.3 for different formulas for sensitivity.

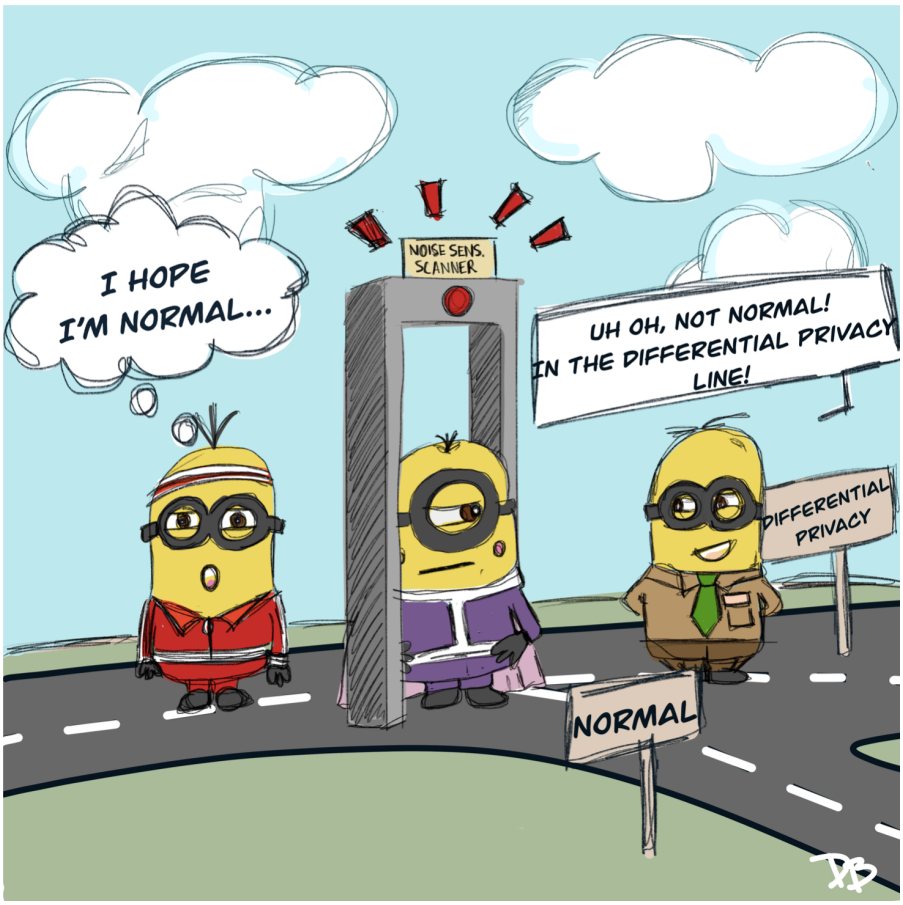

Figure 5.5: Drawing: MNM
This drawing, done by the author, illustrates the concept of MNM through Minions passing through a sensor. If the minion is deemed "normal" or below the threshold, only a minimal amount of gaussian noise is added to their information. If the minion sets off the sensor (above the threshold) then more carefully added Laplace noise must be added to them.

Main result. The main result is the following theorem.

Theorem 5.3.2. *Let $\varepsilon > 0$, $\delta \in (0,1)$. The Mixed Noise Mechanism*

$$O_f(\boldsymbol{x}) = f(\boldsymbol{x}) + \mathcal{N}(0, \sigma^2_{MNM})\mathbb{1}\{\Delta f \leq t_0\} + Lap(\Delta f/\varepsilon)\mathbb{1}\{\Delta f > t_0\}\,,$$

with the threshold $t_0 = \sqrt{\pi}\delta\epsilon/2$ is (ε, δ)-differentially private.

Proof of Theorem 5.3.2. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be neighbouring datasets. Without loss of generality assume that $f(\boldsymbol{x}) = 0$ which implies $f(\boldsymbol{y}) = \Delta f$. Note that for two discrete random

variables U and V, and a deterministic set A we can write

$$\begin{aligned}\mathbb{P}(U\mathbb{1}_A + V\mathbb{1}_{A^c} = z) &= \mathbb{P}(U\mathbb{1}_A = z) + \mathbb{P}(V\mathbb{1}_{A^c} = z)\\ &= \mathbb{1}_A\mathbb{P}(U = z) + \mathbb{1}_{A^c}\mathbb{P}(V = z).\end{aligned}$$

The same applies when densities are used. In what follows, we will write $g(z; Z)$ to denote a density of a random variable Z at point z. We will write for short $\mathcal{N} = \mathcal{N}(0, \sigma^2)$ and $\mathrm{Lap} = \mathrm{Lap}(b)$ with $b = \Delta f/\varepsilon$. Also, for $a, b, c, d > 0$ we have, $\dfrac{a+b}{c+d} < \dfrac{a}{c} + \dfrac{b}{d}$. Using these two facts we can write,

$$\begin{aligned}&\frac{g(z; \mathcal{N}\mathbb{1}\{\Delta f < t_0\} + \mathrm{Lap}\mathbb{1}\{\Delta f \geq t_0\})}{g(z; f(D') + \mathcal{N}\mathbb{1}\{\Delta f < t_0\} + \mathrm{Lap}\mathbb{1}\{\Delta f \geq t_0\})}\\ &= \frac{g(z; \mathcal{N}\mathbb{1}\{\Delta f < t_0\}) + g(z; \mathrm{Lap}\mathbb{1}\{\Delta f \geq t_0\})}{g(z - \Delta f; \mathcal{N}\mathbb{1}\{\Delta f < t_0\}) + g(z - \Delta f; \mathrm{Lap}\mathbb{1}\{\Delta f \geq t_0\})}\\ &\leq \frac{g(z; \mathcal{N}\mathbb{1}\{\Delta f < t_0\})}{g(z - \Delta f; \mathcal{N}\mathbb{1}\{\Delta f < t_0\})} + \frac{g(z; \mathrm{Lap}\mathbb{1}\{\Delta f \geq t_0\})}{g(z - \Delta f; \mathrm{Lap}\mathbb{1}\{\Delta f \geq t_0\})}\\ &= \mathbb{1}\{\Delta f < t_0\}\frac{g(z; \mathcal{N})}{g(z - \Delta f; \mathcal{N})} + \mathbb{1}\{\Delta f \geq t_0\}\frac{g(z; \mathrm{Lap})}{g(z - \Delta f; \mathrm{Lap})}\end{aligned}$$

The last part is bounded by $\exp(\varepsilon)$. The bound for the first part follows from Lemma 5.3.3 below. □

Lemma 5.3.3. *Let $\delta \in (0, 1)$, $\varepsilon > 0$, $\sigma > 0$ be fixed. If*

$$0 < \Delta f < \sqrt{\frac{2\sqrt{2}\sigma^3}{\sqrt{\pi}\delta} + 2\sigma^2\epsilon} - \sqrt{\frac{2\sqrt{2}\sigma^3}{\sqrt{\pi}\delta}} \tag{5.15}$$

then

$$g(z; \mathcal{N}) \leq e^{\varepsilon} g(z - \Delta f; \mathcal{N}) + \delta\ .$$

Proof. We will write for short $\mathcal{N} = \mathcal{N}(0, \sigma^2)$. Then,

$$\frac{g(z; \mathcal{N})}{g(z - \Delta f; \mathcal{N})} = e^{1/2\sigma^2|2z\Delta f + \Delta^2(f)|}\ .$$

This expression does not exceed e^{ε} whenever $z \leq \frac{\sigma^2\varepsilon}{\Delta f} - \frac{\Delta f}{2}$.

Let

$$t = t(\Delta f) = \frac{\sigma^2\varepsilon}{\Delta f} - \frac{\Delta f}{2}\ . \tag{5.16}$$

Since we are interested in small values of Δf, we can assume that t is nonnegative. We use the following classical bound on the tail of the normal distribution.

$$\mathbb{P}(\mathcal{N} > t) \leq \frac{\sigma}{\sqrt{2\pi}} e^{-t^2/(2\sigma^2)} .$$

Let $u > 0$. Then the Taylor expansion of the exponential function gives

$$e^{-u} = \frac{1}{e^u} = \frac{1}{1 + u + \frac{u^2}{2!} + \frac{u^3}{3!} + \cdots} \leq \frac{1}{u} . \tag{5.17}$$

Choosing $u = t^2/(2\sigma^2)$ we obtain

$$\mathbb{P}(\mathcal{N} > t) \leq \frac{\sigma}{\sqrt{2\pi}} \frac{1}{e^{t^2/2\sigma^2}} \leq \frac{\sigma^3\sqrt{2}}{\sqrt{\pi}} t^{-2} .$$

In order to obtain (ε, δ)-differential privacy, we need $\mathbb{P}(\mathcal{N} > t) \leq \delta/2$. This amounts to solving the following inequality

$$\frac{2\sqrt{2}}{\sqrt{\pi}} \frac{\sigma^3}{\delta} < \left(\frac{\sigma^2 \varepsilon}{\Delta f} - \frac{\Delta f}{2} \right)^2 . \tag{5.18}$$

This gives a quadratic inequality with Δf as a variable, yielding (5.15).Recalling that the expression in the bracket on the right hand side is positive, we can rewrite this

$$\frac{(\Delta f)^2}{2} + \Delta f \sqrt{\frac{2\sqrt{2}\sigma^3}{\sqrt{\pi}\delta}} - \sigma^2 \varepsilon < 0 . \tag{5.19}$$

When solving the quadratic equation and keeping in mind that Δf is positive, we get

$$0 < \Delta f < \sqrt{\frac{2\sqrt{2}\sigma^3}{\sqrt{\pi}\delta} + 2\sigma^2\epsilon} - \sqrt{\frac{2\sqrt{2}\sigma^3}{\sqrt{\pi}\delta}} . \tag{5.20}$$

□

Remark 5.3.4. Since Δf is small, we can omit the term $\frac{\Delta f}{2}$ in the inequality (5.18). This leads to

$$\Delta f \leq \frac{\varepsilon \delta^{1/2} \sigma^{1/2} \pi^{1/4}}{2^{3/4}} .$$

If $\sigma^2 = 2(\Delta f/\varepsilon)^2$, then the bound becomes

$$\Delta f \leq \frac{\pi^{1/2} \delta \varepsilon}{2} .$$

This yields the threshold in (5.20).

Remark 5.3.5. The threshold can be further refined by keeping the term u^q in (5.17) for some positive integer q.

Application of MNM. We present the algorithm for the implementation of MNM [5.3.6] with $f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i/n =: \bar{\boldsymbol{x}}$ and $\Delta f = \Lambda/n$. The implementation does not change when choosing a different f and subsequently use a different Δf when applicable.

Algorithm 5.3.6. [Mixed Noise Mechanism]
Input: $(x_1, ..., x_n), n \geq 0$
Output: $O_f(\boldsymbol{x}) = f(\boldsymbol{x}) + Z$
Fix $\varepsilon > 0$ and $\delta \in (0, 1)$
Choose $f(\boldsymbol{x}) = \bar{\boldsymbol{x}}$
Calculate Δf
Fix $\sigma^2_{MNM} = 2(\Delta f/\varepsilon)^2$
Set threshold $t_0 = \dfrac{\pi^{1/2}\delta\varepsilon}{2}$
If $\Delta f > t_0$ $O_f(\boldsymbol{x}) = f(\boldsymbol{x}) + \text{Lap}(\Delta f/\varepsilon)$
If $\Delta f \leq t_0$ $O_f(\boldsymbol{x}) = f(\boldsymbol{x}) + \text{N}(0, \sigma^2_{MNM})$

Experimental Analysis. With the theoretical justification provided in the sections above, we further tested MNM in a query-based setting to evaluate how well it works in practice. In this way, we can know the impact it has on real datasets to improve data utility while meeting the definition of differential privacy. The experiments in this section were conducted using a public dataset to improve data protection and privacy.

The summary of our analysis follows:

- The MNM and Laplace mechanisms perform similarly from the point of view of data utility.
- Both the MNM and Laplace mechanisms outperform the Gaussian mechanism.

Measures of data utility

The term "data utility" is inherently subjective, and for the purposes of this section, we will assess data utility in a few ways. As always, we refer to data utility as how similar or close the transformed statistic is to the true statistic. This is accomplished by analyzing the amount of noise added over several runs of an experiment and taking a ratio to look at the behaviour of MNM versus differential privacy and approximate differential privacy. Finally, an analysis was conducted around the 99% confidence interval for the true mean, with 1000 iterations over different distributional parameters and varying values of δ. We consider the data utility to be high if the percentage of estimators lying within the confidence interval is high. The results of these are summarized in Table 5.1.

Sample Mean

To show the effects of MNM versus other methods, we tested the mechanism on the sample mean estimator when data x come from four different underlying distributions: Normal, Exponential, Pareto and Student-t. Normal and exponential distributions are widely used in practice, while Pareto and Student-t are often used to model data with fat tails. We note that these distribution have infinite support. Hence, the global sensitivity is ∞. As such, in our experiments we are going to use the local sensitivity. We recall that this may violate differential privacy in all mechanisms studied.

However, our goal is to compare data utility stemming from different randomization mechanisms. We aim to compare the effects of a differentially private mechanism (Laplace noise added), an Approximate differentially private mechanism (Gaussian noise is added), and MNM. We test each mechanism on each underlying distribution with varying parameters. We fix $\varepsilon = 1$ and keep each underlying dataset of size $n = 500$. We also vary the values of $\delta = \frac{1}{n}, \frac{1}{2n}, \frac{1}{n^{3/2}}$, and $\frac{1}{n^2}$. We ran 1000 simulations for each experiment, where the true mean of the underlying dataset was calculated. From there, we calculated a noisy mean for each of the three noise mechanisms and stored these values.

In order to determine the effects of these different mechanisms, we present two types of graphs. First, we sum the squared differences of each noisy mean and the true mean. We then examine the ratios of the Laplace sum and Gaussian sum compared to the ratio of the Laplace sum and MNM sum. If the ratio is less than 1 this indicates that the Laplace mechanism performs better in terms of adding less noise and therefore having higher data utility. If the ratio is greater than 1, this indicates that either the Gaussian or MNM performs better. See e.g. Figure 5.6.

Next, in order to illustrate variability of the noisy mean, we show box-plots for each mechanism. See e.g. Figure 5.7.

Normal. When the underlying data distribution is normal, we can see that the Laplace mechanism tends to perform better than the Gaussian and MNM. There are cases when the MNM ratio is greater than 1, but we can see that it oscillates around 1, indicating that either mechanism can perform well. In terms of variability of the noisy mean, we can see that the approximate differential privacy method produces more outliers. See Figure 5.6 and 5.7.

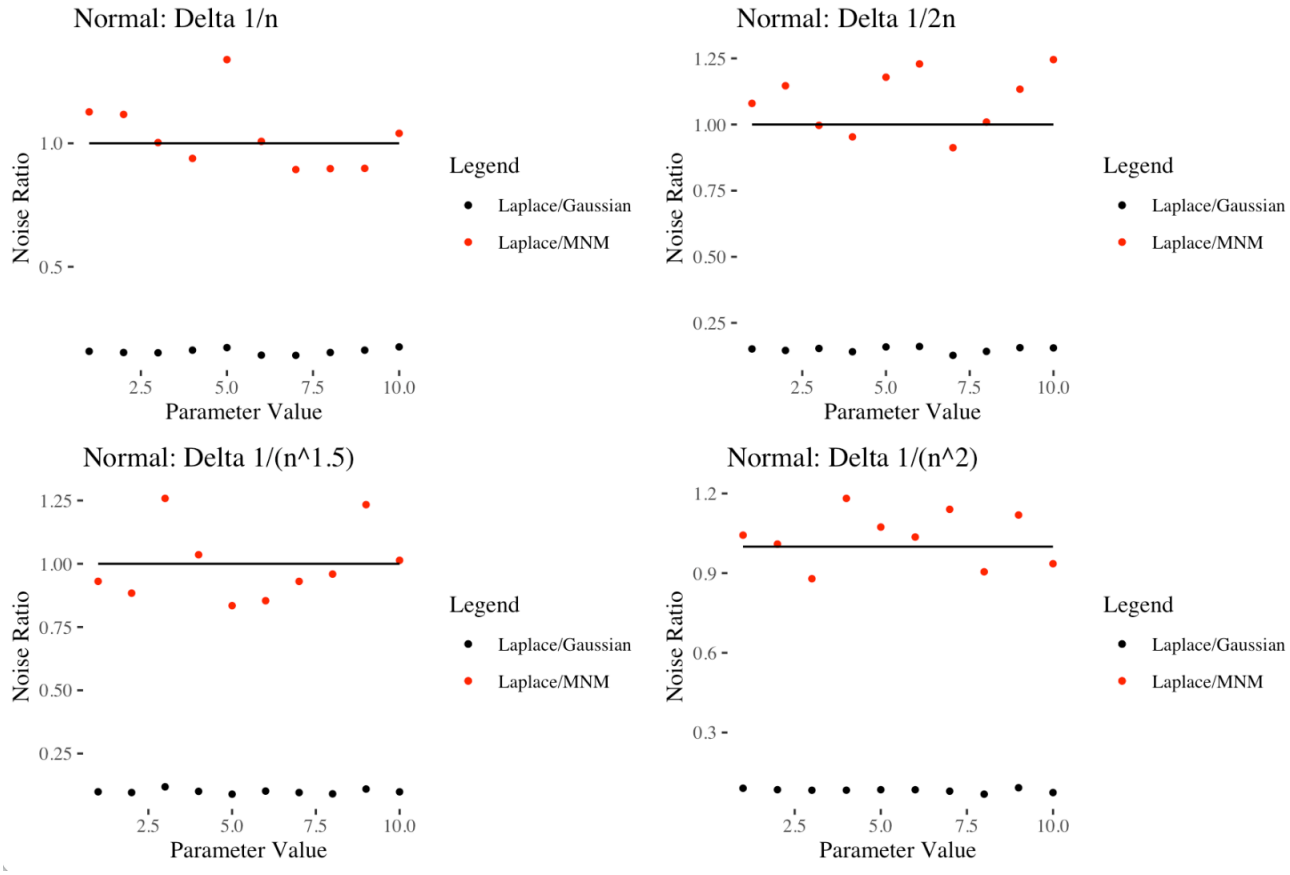

Figure 5.6: MNM results - normal distribution
Values bigger than one indicate that the MNM performs better than Laplace from a data utility point of view. Note further that the Gaussian mechanism has a poor performance from the perspective of data utility. The underlying data distribution is Normal.

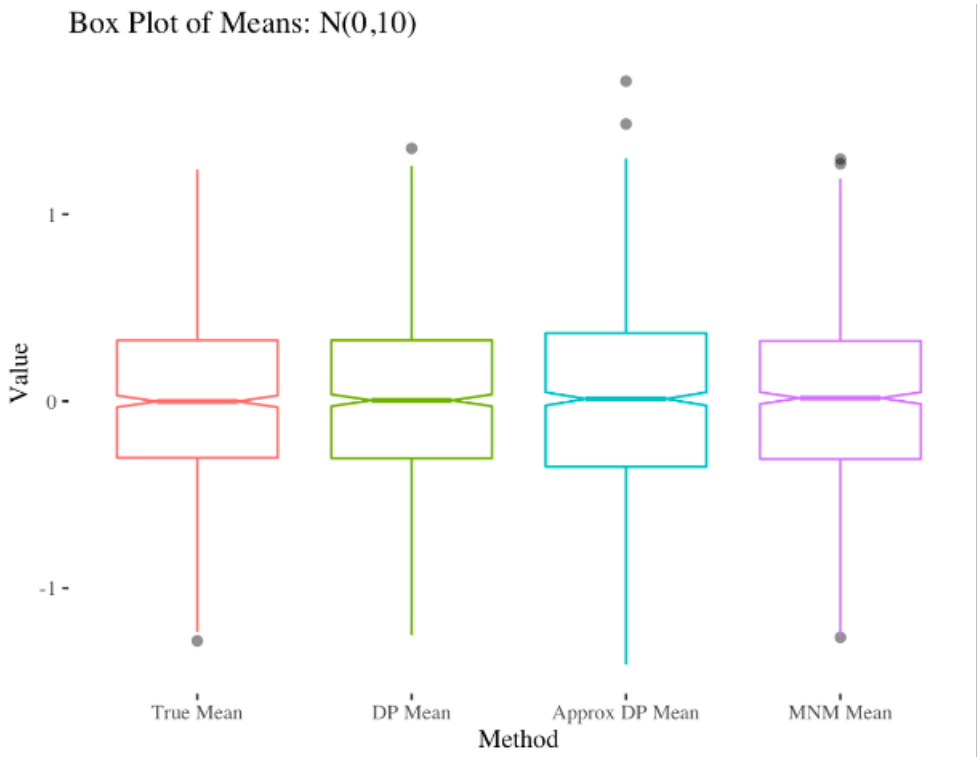

Figure 5.7: MNM - Box plot for the normal
Box Plot of the noisy sample mean when the underlying distribution is $\mathcal{N}(0, 10)$.

Student-t. We see very similar results when the underlying distribution is a Student-t. On average the Laplace mechanism performs well, but there are instances that indicate greater data utility could be achieved when using the MNM mechanism. This is promising in terms of the ease of transparency when adding noise to statistics or datasets. The box plot showed similar results in terms of preserving distributional properties with the MNM mechanism. See Figure 5.8 and 5.9.

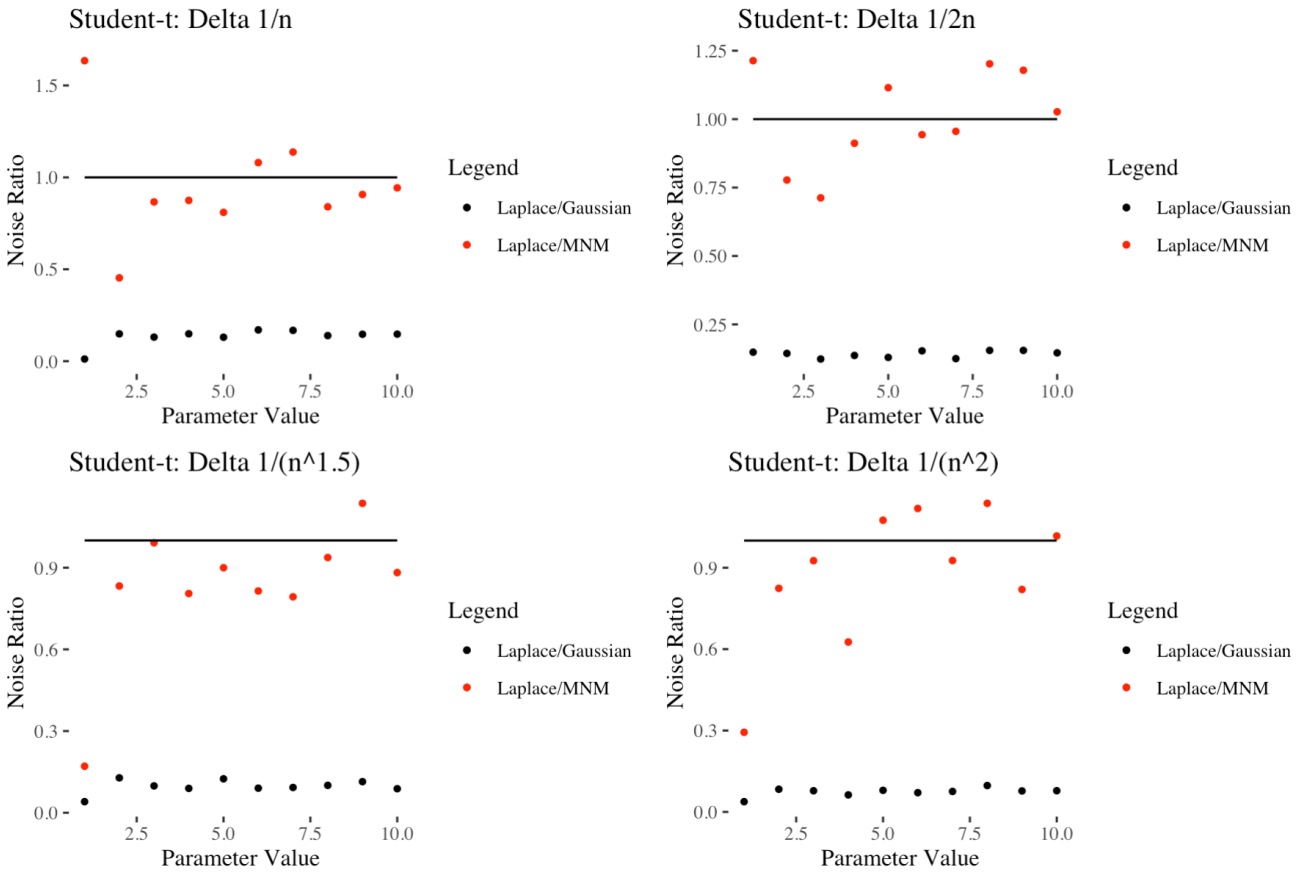

Figure 5.8: MNM results - student-t distribution
Values bigger than one indicate that the MNM performs better than Laplace from a data utility point of view. Note further that the Gaussian mechanism has a poor performance from the perspective of data utility. The underlying data distribution is Student-t.

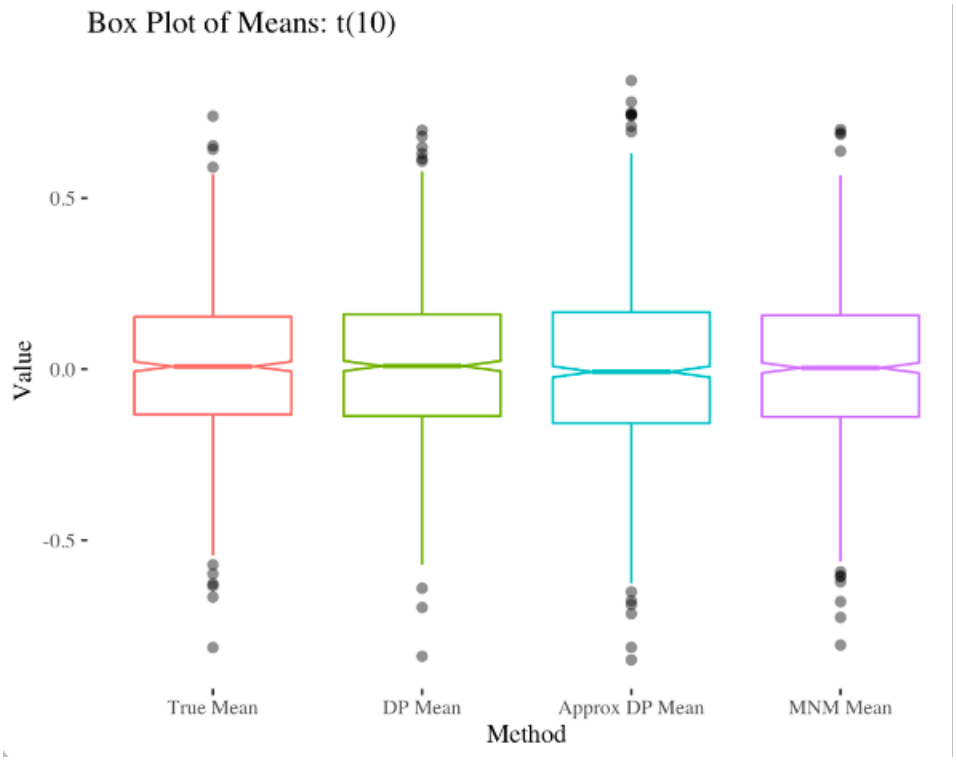

Figure 5.9: MNM - Box plot for the student-t
Box Plot of the noisy sample mean when the underlying distribution is $t(10)$.

Exponential. When the underlying distribution is no longer symmetrical, like the exponential distribution, we start to see some interesting results. MNM, on average, outperformed the Laplace and Gaussian mechanisms. When $\delta = \frac{1}{n}$ the results were quite significant in favour of MNM. This suggests that perhaps MNM performs well in the case of datasets that are one-sided or non-symmetrical. See Figure 5.10 and 5.11.

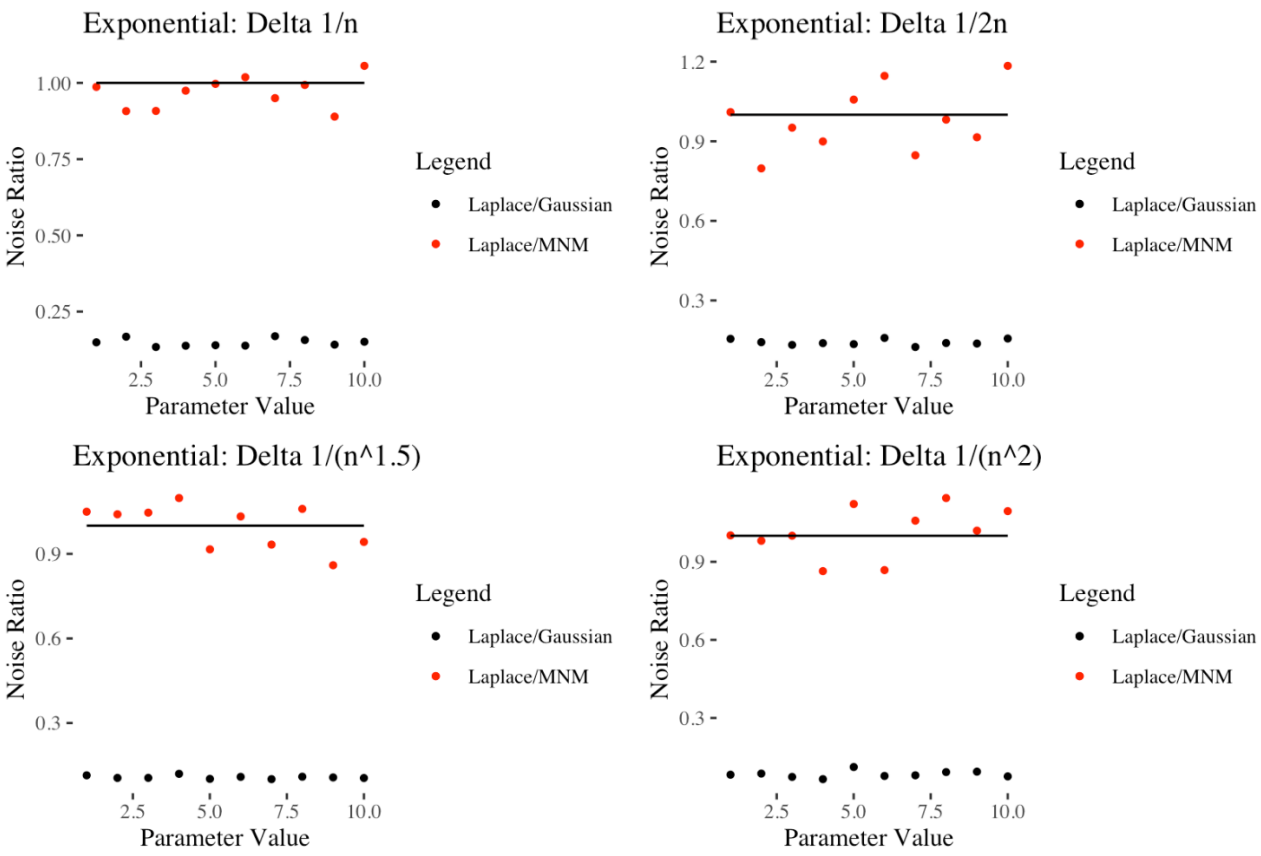

Figure 5.10: MNM results - exponential distribution
Values bigger than one indicate that the MNM performs better than Laplace from a data utility point of view. Note further that MNM performs substantially better compared with the Gaussian mechanism. The underlying data distribution is Exponential.

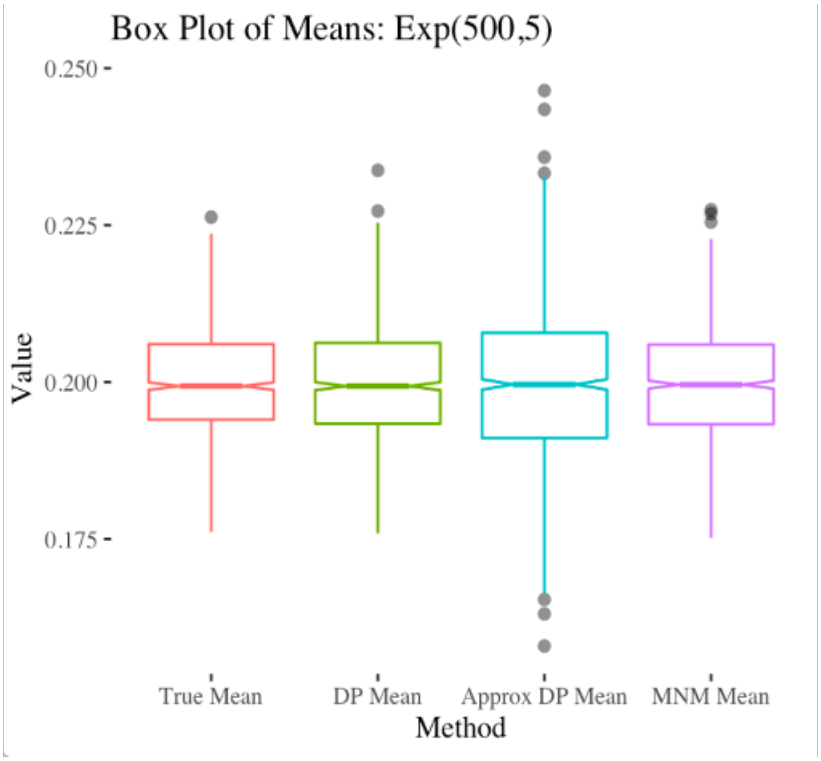

Figure 5.11: MNM - Box plot for the exponential
Box Plot of the noisy sample mean when the underlying distribution is Exp(5).

Pareto. The Pareto distribution is often used in describing scientific, social and natural phenomena. It is skewed with heavy tails, and therefore often used to describe the distribution of income, population and as previously mention stock prices. With many real world applications, it is of interest to see if MNM on Pareto/heavy tailed data can provide an improvement on data utility. See Figure 5.12 and 5.13.

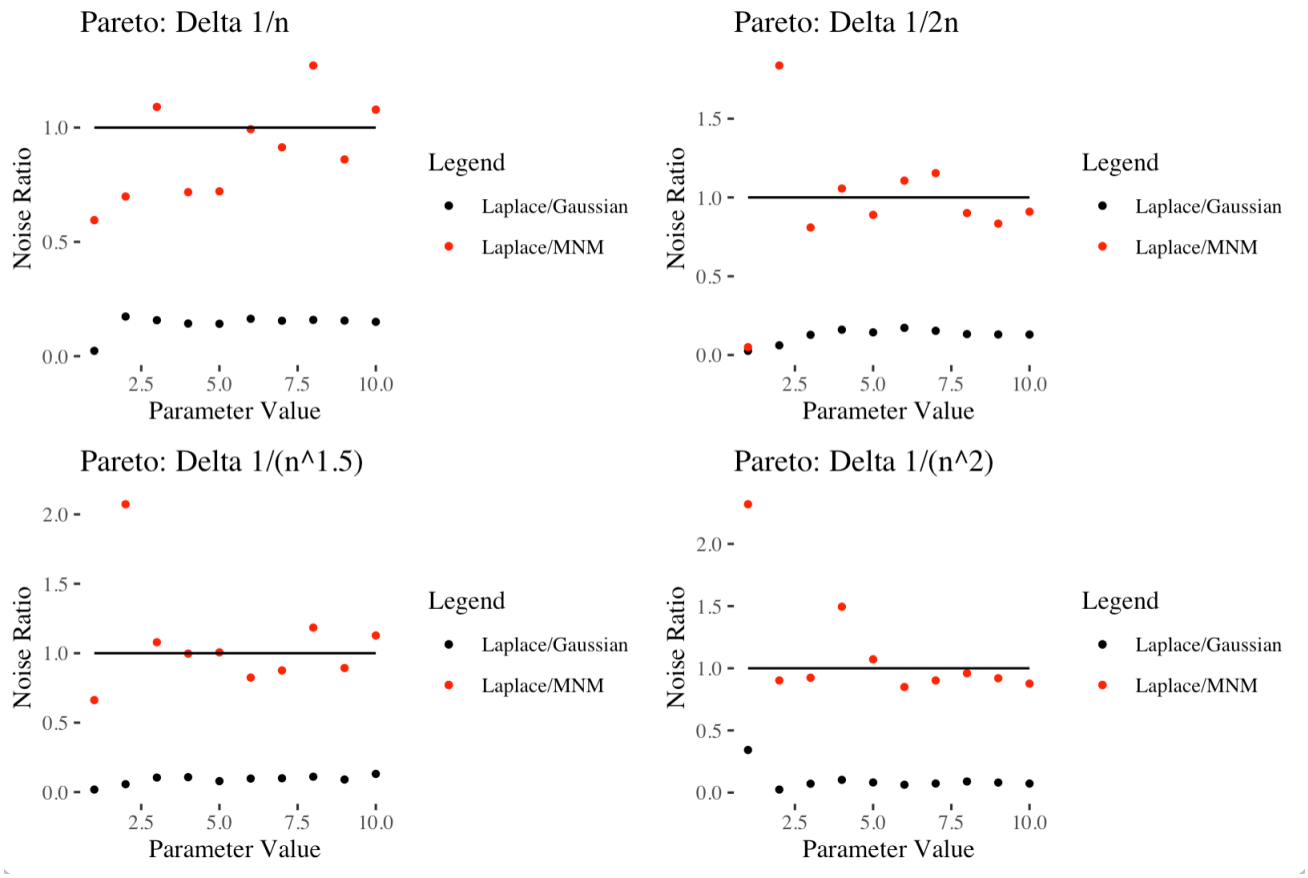

Figure 5.12: MNM results - Pareto distribution
Values bigger than one indicate that the MNM performs better than Laplace from a data utility point of view. Note that the Gaussian mechanism has a poor performance from the perspective of data utility. the underlying data distribution is Pareto.

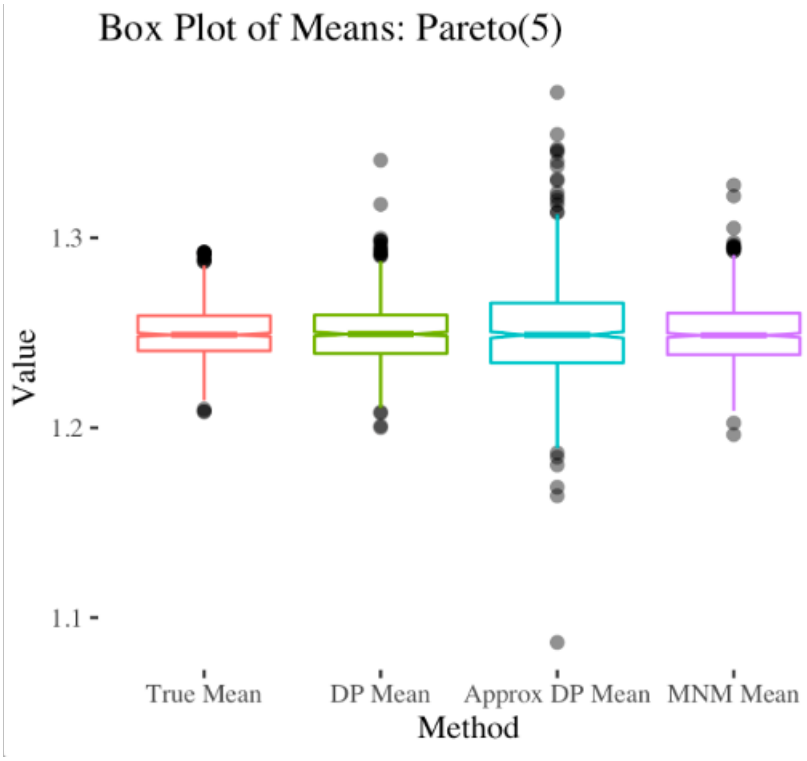

Figure 5.13: MNM - Box plot for the Pareto
Box Plot of the noisy sample mean when the underlying distribution is Pareto(5).

Confidence Interval for the Mean. The purpose of this experiment is to analyze how often the noisy means falls within a 99% confidence interval of the true mean. We simulated datasets varying different parameters for each distribution. For each iteration, the 99% was calculated for the true mean. We created 3 new estimators, differentially private (with Laplace noise added), approximately differentially private (with a Gaussian noise) and MNM. From there we checked for each estimator if it lied within the confidence interval for the mean. If it did, we assigned the value 1 and 0 otherwise. We repeated this experiment 1000 times and calculated the percentage for which the noisy estimators lied within the confidence interval for the mean. The results of this experiment are in Table 5.1.

Median. We also examine another low sensitivity query. We illustrated the smallest version of previous δ values used, $\delta = \frac{1}{n^2}$ for the experimental results. We see much larger ratio values as in comparison to the ratio values of the mean estimator. This is simply due to the fact that the sensitivity of the median is much lower than the sensitivity of the mean. The results for each distribution tested: Exponential, Pareto, Normal, and Student-t are in Figure 5.14. We can see that the ratio values are much higher when

	Laplace DP			Approx DP			MNM		
	$\delta = \frac{1}{n}$	$\delta = \frac{1}{n^{1.5}}$	$\delta = \frac{1}{n^2}$	$\delta = \frac{1}{n}$	$\delta = \frac{1}{n^{1.5}}$	$\delta = \frac{1}{n^2}$	$\delta = \frac{1}{n}$	$\delta = \frac{1}{n^{1.5}}$	$\delta = \frac{1}{n^2}$
Pareto									
$\alpha = 2$	96.8%	95.2%	96.2%	66.5%	56.3%	50.5%	100%	95.5%	96.0%
$\alpha = 5$	98.9%	99.5%	99.0%	82.9%	77.9%	70.9%	99.2%	99.2%	98.5%
$\alpha = 10$	99.7%	99.5%	99.7%	89.9%	85.8%	78.5%	99.8%	99.6%	99.7%
Exponential									
$\lambda = 1$	99.8%	100%	99.9%	95.4%	92.6%	86.6%	100%	99.9%	100%
$\lambda = 5$	100%	99.9%	100%	96.6%	91.5%	87.1%	99.9%	99.9%	99.9%
$\lambda = 10$	99.9%	99.9%	99.8%	95.5%	90.1%	94.2%	99.8%	100%	99.8%
Normal									
$N(0, 1)$	100%	100%	100%	99.7%	99.7%	99.3%	100%	100%	100%
$N(0, 5)$	100%	100%	100%	100%	99.7%	98.5%	100%	100%	100%
$N(0, 10)$	100%	100%	100%	100%	99.8%	99.3%	100%	100%	100%
Student-T									
$t(1)$	91.1%	90.7%	90.0%	51.9%	48.1%	39.1%	91.5%	93.8%	93.7%
$t(5)$	99.9%	100%	100%	96.8%	92.7%	89.2%	99.9%	99.9%	99.8%
$t(10)$	100%	100%	100%	99.7%	97.8%	96.4%	100%	100%	100%

Table 5.1: MNM confidence intervals for various distributions
This table demonstrates the percentage that each sample mean estimator lied within a 99% C.I for the true mean. The experiment was repeated 1000 times for differing values of δ.

they are above 1 and happen more frequently.

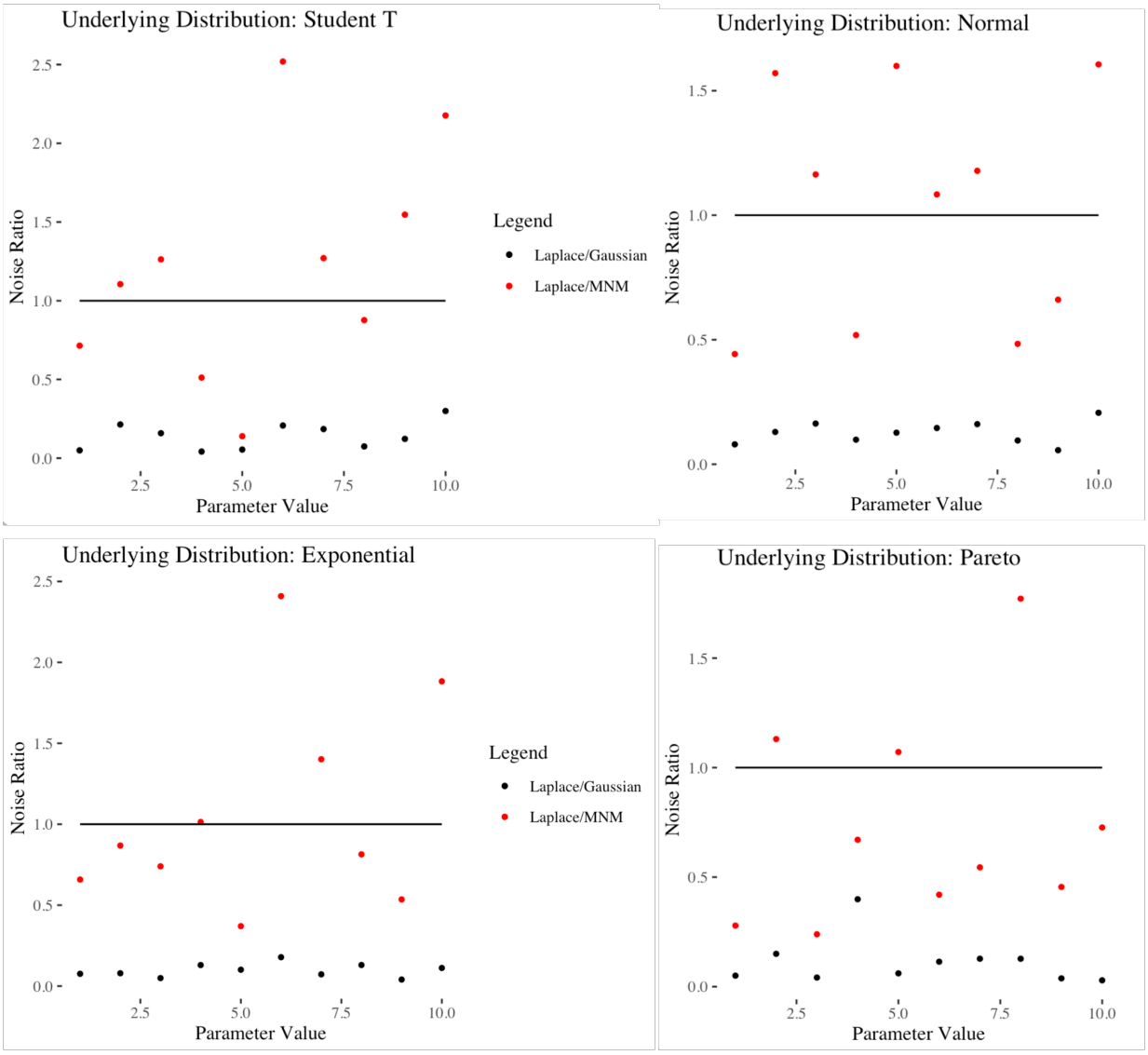

Figure 5.14: MNM results for the median
Values bigger than one indicate that the MNM performs better than the Laplace mechanism (from a data utility point of view) for the median estimator.

We can conclude that in the case of the median, which is indeed a low sensitivity estimator, we see a great improvement in data utility in comparison to existing differentially private mechanisms whilst also maintaining approximate differential privacy.

Conclusion. We studied the differences in ε-differential privacy and (ε, δ)-differential privacy to closely evaluate the implementation and use of a Gaussian noise mechanism for data perturbation. The use of Gaussian noise as the primary method of data perturbation facilitates a stronger connection to foundational theory in statistical inference,

thereby enhancing the utility of perturbed data. Furthermore, while the balance of disclosure risk and data or statistical utility is well understood, this work demonstrates that it is possible to enhance the utility of data without increasing the risk of disclosure.

In particular the discovery of the sensitivity function and the development of a threshold on the noise mechanism facilitated the formulation of a novel approach to introducing Gaussian noise when the data appears "normal" and Laplace noise when violations in the data occur. It was discovered that the Laplace noise can be added when the sensitivity is large, thereby yielding ε-differential privacy. In the case of a low sensitivity value, a normal noise is added, thereby yielding (ε, δ)-differential privacy. The combination of these two techniques results in a mixed noise mechanism (MNM) that markedly enhances data utility while maintaining the same level of privacy as the classical Laplace and Gaussian mechanisms. MNM implements different differentially private noises based on a fixed threshold for the chosen statistical estimator. Given the threshold, it can be determined whether Laplace noise is even needed in order to preserve differential privacy. Finally, the use of MNM allows for the recovery of noisy confidence intervals, the estimation of sensitivity without the use of the privacy budget, and the learning of popular statistical inferences from queries would otherwise be difficult to calculate.

5.4 Blocking

It is a well-established principle that averaging has the effect of reducing the variability in a database. This concept is the foundation for different blocking methods. We start with a novel blocking algorithm (based on the original author's work) and proceed to present its theoretical properties. We present the algorithm introduced by [42]. It is important to note that there are several key distinctions between the two approaches, which will be discussed in greater detail below. Other blocking algorithms are described in the literature; for example, see [33].

5.4.1 Algorithm Block-DP I

Algorithm 5.4.1. 1. Assume that the original population $\mathcal{P}$ has the range $[0, \Lambda]$.

2. Divide $[0, \Lambda]$ into subintervals

$$B_i = \left[(i-1)\frac{\Lambda}{m}, i\frac{\Lambda}{m}\right], \quad i = 1, \ldots, m\ .$$

3. Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be a sample from $\mathcal{P}$.

4. Let $I_i = \{j : x_j \in B_i\}$, $i = 1, \ldots, m$ and $N_i = \sum_{j=1}^{n} \mathbb{1}\{x_j \in B_i\}$ be the number of points in B_i.

5. Set $\boldsymbol{x}_{I_j} = (\boldsymbol{x}_j, j \in I_i)$.

6. Evaluate Δf_i, the global sensitivity of the query f for the population $\mathcal{P}$ restricted to block B_i. It is possible that Δf_i depends on the database $\boldsymbol{x}$, hence we write $\Delta f_i(\boldsymbol{x})$. Assume that there exists $\beta > 0$ and database independent $\Delta\widetilde{f}_i$ such that for each $i = 1, \ldots, m$,

$$\left| \ln \left(\frac{\Delta f_i(\boldsymbol{x})}{\Delta \widetilde{f}_i} \right) \right| < \beta \, . \tag{5.21}$$

7. If $\Delta f_i(\boldsymbol{x}) = \Delta f_i$ does not depend on $\boldsymbol{x}$, release:

$$M(\boldsymbol{X}, (Z_1, \ldots, Z_m)) = h(O_f(\boldsymbol{x}_{I_1}, Z_1), \ldots, O_f(\boldsymbol{x}_{I_m}, Z_m)) \, .$$

   where $O_f(\boldsymbol{x}_{I_i}, Z_i) = f(\boldsymbol{x}_{I_i}) + Z_i$, $Z_i \sim \text{Lap}(\Delta f_i/\varepsilon)$ and h is the appropriate function $h : \mathbb{R}^m \to \mathbb{R}$.

8. If $\Delta f_i(\boldsymbol{x})$ does depend on $\boldsymbol{x}$, release:

$$M(\boldsymbol{X}, (Z_1, \ldots, Z_m)) = h(O_f(\boldsymbol{x}_{I_1}, Z_1), \ldots, O_f(\boldsymbol{x}_{I_m}, Z_m)) \, .$$

   where $O_f(\boldsymbol{x}_{I_i}, Z_i) = f(\boldsymbol{x}_{I_i}) + Z_i$, $Z_i \sim \text{Lap}(2\Delta\widetilde{f}_i/\varepsilon)$ and h is the appropriate function $h : \mathbb{R}^m \to \mathbb{R}$.

The main principle underlying the block method is as follows. While the original sensitivity may be considerable (owing to the range $[0, \Lambda]$) the block sensitivities will be significantly smaller (proportional to Λ/m).

Theorem 5.4.2. *Assume that $\Delta f_i(\boldsymbol{x})$ does not depend on the database $\boldsymbol{x}$. Then the Algorithm Block-DP I is ε-differentially private.*

Proof. Assume first that Δf_i is independent from the data. For $\Delta f_i = \Delta\widetilde{f}_i$, the release $O_f(\boldsymbol{x}_{I_i}, Z_i)$ is ε-differentially private by Theorem 4.3.1. Since $\boldsymbol{x}_{I_1}, \ldots, \boldsymbol{x}_{I_m}$ are disjoint, the release

$$(O_f(\boldsymbol{x}_{I_1}, Z_1), \ldots, O_f(\boldsymbol{x}_{I_m}, Z_m))$$

is also ε-differentially private by Lemma 4.5.14.
Recall by Lemma 4.5.4, that the release

$$M(\boldsymbol{x}, (Z_1, \ldots, Z_m))$$

is ε-differentially private. □

As previously stated in DP Fallacy 5.1.1, data-dependent local sensitivity may lead to a violation of differential privacy. However, in our algorithm, the block sensitivity $\Delta f_i(\boldsymbol{x})$ is data dependent. Nevertheless, we have managed to achieve approximate differential privacy.

Theorem 5.4.3. *The Algorithm Block-DP I is (ε, δ)-DP with*

$$\delta = \delta(\varepsilon, \beta) = e^{\varepsilon/2} e^{-\varepsilon/(2\beta)} ,$$

where β is defined in (5.21).

Proof. It is sufficient to demonstrate that the release $O_f(\boldsymbol{x}_{I_i}, Z_i)$ is (ε, δ)-differentially private. The proof follows the general approach presented in Section 5.2.2. For the sake of simplicity in notation, we write $\boldsymbol{x}_i = \boldsymbol{x}_{I_i}$. Let $\boldsymbol{y}$ be the neighbour of $\boldsymbol{x}$ and set $\boldsymbol{y}_i = (\boldsymbol{y}_j, j \in I_i)$. It should be noted that only one of the $\boldsymbol{y}_i$, $i = 1, \ldots, m$, differs from $\boldsymbol{x}_i$, with all other $\boldsymbol{y}_i = \boldsymbol{x}_i$ (there is only one entry in the entire database $\boldsymbol{y}$ that is different). Recalling the result of Lemma 2.2.3, we can write $Z_i = (2/\varepsilon)(\Delta\widetilde{f}_i)Z'$, where Z' is Laplace(1).

Then

$$\mathbb{P}(f(\boldsymbol{x}_i) + (2/\varepsilon)(\Delta\widetilde{f}_i)Z' \in B) = \frac{1}{2}\int_B e^{-\frac{|q-f(\boldsymbol{x}_i)|}{\Delta\widetilde{f}_i}} dq$$
$$= \frac{1}{2}\int_B \underbrace{e^{\frac{-|q-f(\boldsymbol{x}_i)|+|q-f(\boldsymbol{y}_i)|}{\Delta\widetilde{f}_i}(\varepsilon/2)}}_{=:I} e^{-\frac{|q-f(\boldsymbol{y}_i)|}{\Delta\widetilde{f}_i}(\varepsilon/2)} dq.$$

As in the proof of differential privacy, the absolute value of the first part is bounded by

$$|\ln I| \le (\varepsilon/2)\left|\frac{|q-f(\boldsymbol{x}_i)| - |q-f(\boldsymbol{y}_i)|}{\Delta\widetilde{f}_i}\right| \le (\varepsilon/2)\frac{|f(\boldsymbol{x}_i) - f(\boldsymbol{y}_i)|}{\Delta\widetilde{f}_i}$$
$$\le (\varepsilon/2)\underbrace{\frac{|f(\boldsymbol{x}_i) - f(\boldsymbol{y}_i)|}{\Delta f_i(\boldsymbol{x})}}_{\le 1}\frac{\Delta f_i(\boldsymbol{x})}{\Delta\widetilde{f}_i}.$$

Thus

$$\mathbb{P}(f(\boldsymbol{x}_i) + (\varepsilon/2)(\Delta\widetilde{f}_i)Z' \in B) \le e^{\varepsilon/2}\frac{1}{2}\int_B e^{-(\varepsilon/2)\frac{|q-f(\boldsymbol{y}_i)|}{\Delta\widetilde{f}_i}\frac{\Delta f_i(\boldsymbol{x})}{\Delta\widetilde{f}_i}} dq .$$

Finally, as in the proof of Theorem 5.2.17, the assumption (5.21) yields the bound

$$\mathbb{P}(f(\boldsymbol{x}_i) + (\varepsilon/2)(\Delta\widetilde{f}_i)Z' \in B) \le e^{\varepsilon}\mathbb{P}(f(\boldsymbol{y}_i) + (\varepsilon/2)(\Delta\widetilde{f}_i)Z' \in B) + \delta$$

with

$$\delta = \delta(\varepsilon, \beta) = e^{\varepsilon/2} e^{-\varepsilon/(2\beta)} \ ;$$

cf. (5.10). □

The following examples demonstrate the potential for the blocking algorithm to improve data utility.

Example 5.4.4. Assume that the range of the univariate population is $[0, \Lambda]$. Let $f(\boldsymbol{x})$ be the mean query. Then

$$f(\boldsymbol{x}_{I_i}) = \frac{1}{N_i} \sum_{j \in I_i} x_j, \quad \Delta f_i(\boldsymbol{x}) = \frac{\Lambda}{m N_i} \ .$$

Here, Λ/m is the range on block B_i, while N_i is the number of observations in the block. The issue arises due to N_i being dependent on the sample. According to the Law of Large Numbers,

$$N_i \sim \frac{n}{m} \quad \text{as} \quad \text{n} \to \infty \ ,$$

thus $\Delta \widetilde{f}_i = \Lambda/n$, which is the same as Δf.

Take $h(y_1, \ldots, y_m) = \frac{1}{m} \sum_{j=1}^m y_j$, then the release is

$$\frac{1}{m} \left(\frac{1}{N_1} \sum_{j \in I_1} x_j + \cdots + \frac{1}{N_m} \sum_{j \in I_m} x_j \right) + \frac{1}{m} \sum_{i=1}^m Z_i =: \boldsymbol{x}^* + \frac{1}{m} \sum_{i=1}^m Z_i, \quad Z_i \sim \text{Lap}(2/n\varepsilon) \ .$$

Data utility. It can be shown that $\text{Var}\left(\frac{1}{m} \sum_{i=1}^m Z_i\right) = \frac{8}{m} \cdot \frac{\Lambda^2}{n^2 \varepsilon^2}$ is typically smaller than (5.1). Furthermore, it can be demonstrated that $\boldsymbol{x}^*$ yields a finite sample bias as opposed to the sample mean $\boldsymbol{x}$ obtained without blocking.

Privacy. For a sufficiently large value of n, the Law Large Numbers allows for the choice of β to be made close to 0. For a fixed dataset, we can only apply the blocking algorithm if all cells I_j have a sufficient number of observations.

Summary. The blocking method for the mean query may not be of a great advantage over non-blocking.

Example 5.4.5. Assume that the range of the univariate population is $[0, \Lambda]$. Let $f(\boldsymbol{x})$ be the median query. In this example, the global sensitivity is $\Delta f = \Lambda$. We have

$$f(\boldsymbol{x}_{I_i}) = \text{median}(x_j, j \in I_i) \ .$$

Here $\Delta f_i = \Lambda/m$, which does not depend on the sample. Take $h(y_1, \ldots, y_m) = \frac{1}{m}\sum_{i=1}^m y_i$, the release is then:

$$\underbrace{\frac{1}{m}\left(\text{median}(\boldsymbol{x}_{I_1}) + \cdots + \text{median}(\boldsymbol{x}_{I_m})\right)}_{\text{T}} + \frac{1}{m}\sum_{i=1}^m Z_i =: \boldsymbol{x}^* + \frac{1}{m}\sum_{i=1}^m Z_i, \quad Z_i \sim \text{Lap}(2\Lambda/m\varepsilon)\ .$$

Then $\text{Var}\left(\frac{1}{m}\sum Z_i\right) = \frac{8}{m}\cdot\left(\frac{\Lambda}{m\varepsilon}\right)^2$, which is usually smaller than $2\left(\Lambda/\varepsilon\right)^2$, which is the variance if no blocking is applied. The price to pay is again $\boldsymbol{x}^* \neq \text{median}(\boldsymbol{x})$, hence there is a bias.

As will be demonstrated below, there is an improvement in terms of data utility for the median query. It seems reasonable to expect a similar improvement for a number of non-linear queries.

5.4.2 Algorithm Block-DP II

The following algorithm was proposed in [42].

Algorithm 5.4.6.

1. Assume that the original population $\mathcal{P}$ is parametrized by $\theta \in \Theta \subseteq \mathbb{R}$. Assume that $\text{diam}(\Theta) = \Lambda_0$
2. Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be a sample from $\mathcal{P}$.
3. Let I_i, $i = 1, \ldots, m$ be disjoint subsets of $\{1, \ldots, n\}$ such that $|I_1| = \ldots = |I_m|$ and $I_1 \cup \cdots \cup I_m = \{1, \ldots, n\}$.
4. Set $\boldsymbol{x}_{I_j} = (\boldsymbol{x}_j, j \in I_i)$.
5. Let f be an estimator of θ. Release

$$O_f(\boldsymbol{x}, Z) = \frac{1}{m}\sum_{i=1}^m f(\boldsymbol{x}_{I_i}) + Z\ , \tag{5.22}$$

where Z is Laplace$(\Lambda_0/(m\varepsilon))$.

It is first necessary to describe the difference between the two algorithms, Algorithm 5.4.1 and Algorithm 5.4.6:

- Algorithm 5.4.1 is general and can be applied to any query f. On the other hand, in Algorithm 5.4.6 the query f is linked to the original estimation problem (estimation of θ).
- In Algorithm 5.4.1 the blocking restricts the range of the data and hence decreases sensitivity of a query f. In Algorithm 5.4.6 there is no such effect.

- In Algorithm 5.4.1 sensitivity is linked to the range of the population Λ. In Algorithm 5.4.6 sensitivity stems from the range Λ_0 of the parameter θ.

In conclusion, the two blocking methods can be applied in either the same situation or in a completely different setup.

Theorem 5.4.7. *The Algorithm Block-DP II (5.22) is ε-DP.*

Proof. We only need to calculate the sensitivity of

$$g(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m} f(\boldsymbol{x}_{I_i}) \ .$$

Since f is the estimator of θ, and θ has the range Λ_0, it follows that the sensitivity of f is Λ_0. Now, if we have two neighbouring databases $\boldsymbol{x}$ and $\boldsymbol{y}$, since $\boldsymbol{x}_{I_i}$ (resp. $\boldsymbol{y}_{I_i}$) are disjoint, then there is one and only one index i_0 such that $\boldsymbol{x}_{I_{i_0}}$ differs from $\boldsymbol{y}_{I_{i_0}}$. Otherwise, $\boldsymbol{x}_{I_i} = \boldsymbol{y}_{I_i}$ for $i \neq i_0$. Hence the sensitivity of g is Λ_0/m.

□

Numerical experiment. Despite the differences in their configuration, Algorithm 5.4.1 and Algorithm 5.4.6 can be compared from the perspective of data utility in certain scenarios.

Example 5.4.8.

- We generate a database $\boldsymbol{x}$ of $n = 1000$ values from a distribution with a bounded support. We use the uniform distribution on $[0, \Lambda]$. A'priori we have no further information on the range of the median, hence we assume that $\Lambda = \Lambda_0$.

- Then we apply the blocking method I for $m = 5$, $m = 10$, $m = 15$, $m = 20$ and evaluate the medians. We add Laplace noise to each median with $\varepsilon = 1$, and the resulting values are averaged over m.

- Then we apply the blocking method II with $m = 5$, $m = 10$, $m = 15$, $m = 20$ and evaluate the medians. We average over m, and then add a Laplace noise with Laplace$(\Lambda/(m\varepsilon))$.

- We repeat this procedure $N = 1000$ times and calculate the MSE.

The results are displayed in the table below. First, we notice that the MSE values are considerable. Indeed, we keep in mind that $\varepsilon = 1$. It is important to note that in method II, Laplace noise with variance $2(\Lambda/m)^2$ is added. This is large in our case.

	Method I	Method II
$m = 5$	645.21	765.16
$m = 10$	81.69	201.08
$m = 15$	23.83	91.97
$m = 20$	10.45	46.37

Table 5.2: Block DP-I vs Block DP-II
Evaluation of the MSE for Block-DP I and Block-DP II

We note that our blocking method yields much better results. This stems from the fact that we averaged Laplace noises, while method II adds one noise only. However, this comparison is not completely equivalent. Indeed, in Method II we used the worst possible constraint on the median. In a "practical" scenario, it is possible to calculate the confidence interval for the median and utilize it as a constraint. However, since the confidence interval relies on the original dataset, it may potentially lead to issues pertaining to data privacy.

5.5 Bounded Laplace Mechanism

We start with the following example from [27].

Example 5.5.1. Consider a hypothetical scenario in which a census dataset is being queried with the objective of determining the number of individuals born on Mars. The addition of noise from a Laplace mechanism with variance $2/\varepsilon^2$ will satisfy differential privacy. Although the actual number of people born on Mars is zero (at least for the time being), it is necessary to add noise to ensure the privacy of future human martians. Successive outputs from the Laplace mechanism could be: $-1.71, 2.31, -1.20, 0.652$. However bizarre the query, negative outputs are patently illogical and inconsistent. By the symmetry of the Laplace distribution, on average 50% of the outputs will be negative.

Example 5.5.2. For the sake of argument, let us suppose that our interest lies in the noisy standard deviation. Two methods exist for achieving this result, and a bizarre occurrence will be presented as an example. The noisy sample variance (nSV) and the noisy standard deviation (nSD) can be expressed as follows:

$$\text{nSV} = S^2 + \text{Lap}(\Lambda^2/n\varepsilon) \ ,$$

$$\text{nSD} = \sqrt{S^2 + \text{Lap}(\Lambda^2/n\varepsilon)} \ .$$

If we are interested in the nSD, it is possible that the nSV may assume a negative value. In such an instance, it is not feasible to calculate the nSD. If we consider the standard deviation as the direct mechanism, we can achieve the noisy SD through the following mechanism,

$$\text{nSD} = S + \text{Lap}(\Lambda/n\varepsilon) \ .$$

It is obvious that in many situations, the two formulations for the nSD are not the same.

In this section we are interested in queries $Q : \mathcal{D} \times \mathcal{E} \to \text{Dom}$ on datasets $\boldsymbol{x} \in \mathcal{D}$ mapping to a finite domain $\text{Dom} = [l, u] \subset \mathbb{R}$ ($l < u$, both finite). We are concerned only with output perturbation mechanisms, see (2.4):

$$O_f(\boldsymbol{x}, z) = f(\boldsymbol{x}) + z \ , \quad \boldsymbol{x} \in \mathcal{D}, z \in \mathbb{R} \ .$$

Recall that when the noise is independent from the database, (ε, δ)-differential privacy states

$$\mathbb{P}(Q(\boldsymbol{x}, Z) \in B) \leq \mathrm{e}^{\varepsilon}\mathbb{P}(Q(\boldsymbol{y}, Z) \in B) + \delta \ .$$

When the noise is Laplace$(0, \Delta f/\varepsilon)$, pure differential privacy holds. In this case, $Q(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z$, where Z has a Laplace$(0, \Delta f/\varepsilon)$ law. Set $Q_{f(\boldsymbol{x})} := Q(\boldsymbol{x}, Z)$, then the random variable $Q_{f(\boldsymbol{x})}$ has the density

$$g_{Q_{f(\boldsymbol{x})}}(v) = \frac{1}{2b} \exp\left(-\frac{|v - f(\boldsymbol{x})|}{b}\right) \ , \quad b = \frac{\Delta f}{\varepsilon} \ , \quad v \in \mathbb{R}.$$

The idea of the bounded Laplace mechanism is to restrict the domain of the density above to $v \in \text{Dom}$. This will correspond to the randomized query $Q(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z$ being restricted to the domain Dom. This is important to note that the restriction will be on $f(\boldsymbol{x}) + Z$, not on Z. As such, the restriction will depend on the database $\boldsymbol{x}$. This creates some complications.

Definition 5.5.3 (Bounded Laplace Mechanism)**.** *Given $b > 0$ and $\text{Dom} \subset \mathbb{R}$, the bounded Laplace mechanism $Q_{f(\boldsymbol{x})}$ is given by its probability density function:*

$$g_{Q_f(\boldsymbol{x})}(v) = \begin{cases} 0, & \text{if } v \notin D \\ \frac{1}{C_{f(\boldsymbol{x})}} \frac{1}{2b} e^{-\frac{|v - f(\boldsymbol{x})|}{b}}, & \text{if } v \in D, \end{cases}$$

where $C_{f(\boldsymbol{x})} = \int_D \frac{1}{2b} e^{-\frac{|v-f(\boldsymbol{x})|}{b}} dv$ is a normalization constant.

Example 5.5.4. Assume that $\boldsymbol{x}$ describes age. It is reasonable to assume that age is between 0 and 110. Let $f(\boldsymbol{x}) = \min_j x_j$. Assume, the query returns a response of 10. Adding an unbounded Laplace noise (with the real domain) may lead to an unrealistic negative age. Instead, the bounded Laplace mechanism will add Laplace noise with the domain $[-10, 100]$. The resulting output will belong to the prescribed range $[0, 110]$.

In classical differential privacy, we always add Laplace$(0, \Delta f/\varepsilon)$ noise, regardless of the output of the query, $f(\boldsymbol{x})$. Here, the added Laplace noise will depend on the query and hence on the database. As such, the bounded Laplace mechanism does not consistently satisfy differential privacy when utilizing parameters derived from the pure

Laplace mechanism. This is due to the fact that the output is contingent upon the original database (in comparison to the scenario of the Laplace mechanism with local sensitivity). Nevertheless, we can provide an answer when we preserve approximate differential privacy. To this end, it is necessary to note that the data-dependent constant $C_{f(\boldsymbol{x})}$ has the form:

$$C_q := 1 - \frac{1}{2}\left(\exp(-(q-l)/b) + \exp(-(u-q))/b)\right) \ ,$$

where $q = f(\boldsymbol{x})$. Define

$$\Delta C = \frac{C_{l+\Delta f}}{C_l} \ .$$

Unlike the constant $C_{f(\boldsymbol{x})}$, the new constant ΔC does not depend on the database (but it depends on b).

Theorem 5.5.5. *The bounded Laplace mechanism is (ε, δ)- differentially private whenever*

$$b \geq \frac{\Delta f}{\varepsilon - \log \Delta C - \log(1-\delta)}.$$

5.6 Pre-processing vs Post-processing

This chapter presents a comparative analysis of output perturbation mechanisms, O_f, and sanitized response mechanisms, S_f, from the perspective of data utility. It should be recalled that the output perturbation mechanism corresponds to post-processing (adding noise to the query), while the sanitized response mechanism corresponds to pre-processing (adding noise first, then applying query).

We analyze several estimators. It is not feasible to develop a general theory that covers a large class of estimators. The main findings are as follows:

- In general, the same level of data utility can be achieved with less privacy through pre-processing. Conversely, the same level of privacy can be achieved with better data utility through post-processing.
- For the sample mean query pre- and post-processing yield an unbiased estimator of the population mean. However, the pre-processing method results in a reduction in data utility, as measured by the MSE.
- For the sample variance, query pre-processing leads to bias. Furthermore, it leads to a lower data utility, as measured by the MSE.

Sample Mean. Let $\boldsymbol{x} = (x_1, \cdots, x_n)$ be a database. We treat this database as fixed. Assume that the data come from a population with range $[0, \Lambda]$. We are interested in estimating the population mean using the sample mean.

Post-processing

We consider the output perturbation mechanism a

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z,$$

where $f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i/n$ and $Z \sim \text{Lap}(\Delta f/\varepsilon)$. This algorithm is ε-differentially private.

Recall now that $\Delta f = \Lambda/n$. Then

$$\mathbb{E}[O_f(\boldsymbol{x}, Z)] = \mathbb{E}[f(\boldsymbol{x}) + Z] = f(\boldsymbol{x}) \ ,$$
$$\text{Var}(O_f(\boldsymbol{x}, Z)) = \text{Var}(Z) = \frac{2\Lambda^2}{\varepsilon^2 n^2}.$$

So in the case of post-processing, the randomized query $f(\boldsymbol{x}) = \bar{\boldsymbol{x}}$ leads to an unbiased estimator.

Pre-processing

In the case of pre-processing, we define $\boldsymbol{Y} = (x_1 + Z_1, \cdots, x_n + Z_n)$, where Z_i are independent with the distribution Laplace(Λ/ε). Let $\boldsymbol{Z} = (Z_1, \ldots, Z_n)$. We define the sanitized response mechanism as

$$S_f(\boldsymbol{x}, \boldsymbol{Z}) = f(\boldsymbol{x} + \boldsymbol{Z}) = f(\boldsymbol{x}) + f(\boldsymbol{Z}).$$

This algorithm is ε-differentially private. We wish to calculate the expected value and variance of S_f and then compare these results to that of the post-processing results.

$$\mathbb{E}[S_f(\boldsymbol{x}, \boldsymbol{Z})] = f(\boldsymbol{x})$$
$$\text{Var}(S_f(\boldsymbol{x}, \boldsymbol{Z})) = \frac{1}{n}\text{Var}(Z_1) = \frac{2\Lambda^2}{\varepsilon^2 n}.$$

We can conclude that due to the linearity of the sample mean, the expected value of each mechanism O_f, S_f is the same, but there is some difference between the variances. In order to understand the relationship of ε between the post-processed statistic and pre-processed database statistic we set the variances of O_f and S_f equal and solve for epsilon. For this, set

$$\text{Var}(S_f(\boldsymbol{x}, \boldsymbol{Z})) = \frac{2\Lambda^2}{\varepsilon_0^2 n} \ ,$$

where ε_0 is the "epsilon" for the pre-processing case. Then we solve the following equation for ε_0 :

$$\frac{2\Lambda^2}{n\varepsilon_0^2} = \frac{2\Lambda^2}{\varepsilon^2 n^2}$$
$$n\varepsilon_0^2 = n^2\varepsilon^2$$
$$\varepsilon_0 = \sqrt{n}\varepsilon \ .$$

Therefore, in order to achieve equivalent data utility for both methods (data utility is measured by the MSE), it is necessary to utilize pre-processing with much larger "epsilon" than for post-processing. In other words, at a given level of data utility, pre-processing affords less privacy, whereas at a given level of privacy, post-processing yields superior data utility.

Sample Variance. Let $\boldsymbol{x} = (x_1, \cdots, x_n)$ be a database. We treat this database as fixed. Assume that the data come from a population with range $[0, \Lambda]$. We are interested in estimating the population variance using the sample variance.

Post-processing

We consider the output perturbation mechanism

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z,$$

where $f(\boldsymbol{x}) = \frac{1}{n-1}\sum_{i=1}^n (x_i - \bar{x})^2$ and $Z \sim \text{Laplace}(\Delta f/\varepsilon)$. Recall from Example 2.1.4 that the global sensitivity of the variance query is $\Delta f = \Lambda^2/n$. Recall also from Example 5.5.2 that this may lead to unfeasible noisy estimators of the variance. Then

$$\begin{aligned} \mathbb{E}[O_f(\boldsymbol{x}, Z)] &= f(\boldsymbol{x}) \ , \\ \text{Var}(O_f(\boldsymbol{x}, Z)) = \text{Var}(Z) &= 2\left(\frac{\Delta f}{\varepsilon}\right)^2 = 2\frac{\Lambda^4}{\varepsilon^2 n^2} \ . \end{aligned}$$

Pre-processing

In the case of pre-processing, we define $\boldsymbol{Y} = (x_1 + Z_1, \cdots, x_n + Z_n)$, where Z_i are independent with the distribution $\text{Laplace}(\Lambda/\varepsilon)$. It is important to note that the parameter of the Laplace distribution is different as compared to the post-processing. Indeed, in the post-processing case the sensitivity Λ^2/n is stems from the query (sample variance), while in the pre-processing case the sensitivity Λ stems from the range of the database. Let $\boldsymbol{Z} = (Z_1, \ldots, Z_n)$. Then

$$S_f(\boldsymbol{x}, \boldsymbol{Z}) = \frac{1}{n-1}\sum_{i=1}^n (Y_i - \bar{\boldsymbol{Y}})^2 = \frac{1}{n-1}\sum_{i=1}^n (x_i + Z_i - \bar{\boldsymbol{x}} - \bar{\boldsymbol{Z}})^2 \ .$$

Since the random variables Z_i are centered, we have

$$\begin{aligned} \mathbb{E}[S_f(\boldsymbol{x}, \boldsymbol{Z})] &= \frac{1}{n-1}\sum_{i=1}^n (x_i - \bar{\boldsymbol{x}})^2 + \mathbb{E}\left[\frac{1}{n-1}\sum_{i=1}^n (Z_i - \bar{\boldsymbol{Z}})^2\right] \\ &= \frac{1}{n-1}\sum_{i=1}^n (x_i - \bar{\boldsymbol{x}})^2 + \text{Var}(Z_1) = \frac{1}{n-1}\sum_{i=1}^n (x_i - \bar{\boldsymbol{x}})^2 + 2\frac{\Lambda^2}{\varepsilon^2} \ . \end{aligned} \tag{5.23}$$

Thus, even though the sample variable based on the original data is an unbiased estimator of the population variance, the noise sample variance is biased.

Next, the formula for the variance is

$$\mathrm{Var}(S_f(\boldsymbol{x}, \boldsymbol{Z})) = \mathbb{E}\left[S_f^2(\boldsymbol{x}, \boldsymbol{Z})\right] - (\mathbb{E}[S_f(\boldsymbol{x}, \boldsymbol{Z})])^2 \,. \tag{5.24}$$

We calculate the first term on the right hand side of (5.24):

$$\begin{aligned}
\mathbb{E}\left[S_f^2(\boldsymbol{x}, \boldsymbol{Z})\right] &= \frac{1}{(n-1)^2}\mathbb{E}\left(\sum_{i=1}^n (x_i + Z_i - \bar{\boldsymbol{x}} - \bar{\boldsymbol{Z}})^2\right)^2 \\
&= \frac{1}{(n-1)^2}\mathbb{E}\left(\sum_{i=1}^n (x_i - \bar{\boldsymbol{x}})^2 + (Z_i - \bar{\boldsymbol{Z}})^2 + 2(x_i - \bar{\boldsymbol{x}})(Z_i - \bar{\boldsymbol{Z}})\right)^2 \\
&= \frac{1}{(n-1)^2}\mathbb{E}\Bigg(\sum_{i=1}^n (x_i - \bar{\boldsymbol{x}})^4 + (Z_i - \bar{\boldsymbol{Z}})^4 + 4(x_i - \bar{\boldsymbol{x}})^3(Z_i - \bar{\boldsymbol{Z}}) \\
&\quad + 4(x_i - \bar{\boldsymbol{x}})(Z_i - \bar{\boldsymbol{Z}})^3 + 6(x_i - \bar{\boldsymbol{x}})^2(Z_i - \bar{\boldsymbol{Z}})^2\Bigg) \\
&= \frac{1}{(n-1)^2}\sum_{i=1}^n (x_i - \bar{\boldsymbol{x}})^4 + \frac{1}{(n-1)^2}\sum_{i=1}^n \mathbb{E}(Z_i - \bar{\boldsymbol{Z}})^4 + \frac{6}{(n-1)^2}\sum_{i=1}^n (x_i - \bar{\boldsymbol{x}})^2\mathbb{E}(Z_i - \bar{\boldsymbol{Z}})^2 \,.
\end{aligned}$$

Indeed, because the random variables Z_i are symmetric around zero we have

$$\mathbb{E}[(Z_i - \bar{\boldsymbol{Z}})] = \mathbb{E}[Z_i] - \mathbb{E}[\bar{\boldsymbol{Z}}] = 0$$

and

$$\begin{aligned}
\mathbb{E}[(Z_i - \bar{\boldsymbol{Z}})^3] &= \mathbb{E}[Z_i^3] - 3\mathbb{E}[Z_i^2\bar{\boldsymbol{Z}}] + 3\mathbb{E}[Z_i(\bar{\boldsymbol{Z}})^2] - 3\mathbb{E}[(\bar{\boldsymbol{Z}})^3] \\
&= 0 - \frac{3}{n}\sum_{j=1}^n \mathbb{E}[Z_i^2 Z_j] + \frac{1}{n^2}\sum_{j,k=1}^n \mathbb{E}[Z_i Z_j Z_k] - \frac{3}{n^3}\sum_{i,j,k=1}^n \mathbb{E}[Z_i Z_j Z_k] = 0 \,.
\end{aligned}$$

Next, we evaluate

$$\begin{aligned}
\mathbb{E}[(Z_i - \bar{\boldsymbol{Z}})^2] &= \mathrm{Var}(Z_i - \bar{\boldsymbol{Z}}) \\
&= \mathrm{Var}(Z_i) + \mathrm{Var}(\bar{\boldsymbol{Z}}) - 2\mathrm{Cov}(Z_i, \bar{\boldsymbol{Z}}) \\
&= \mathrm{Var}(Z_i) + \mathrm{Var}(\bar{\boldsymbol{Z}}) - 2\frac{1}{n}\sum_{j=1}^n \mathrm{Cov}(Z_i, Z_j) \\
&= \frac{2\Lambda^2}{\varepsilon^2} + \frac{1}{n}\frac{2\Lambda^2}{\varepsilon^2} - \frac{2}{n}\frac{2\Lambda^2}{\varepsilon^2} = \frac{(n-1)}{n}\cdot\frac{2\Lambda^2}{\varepsilon^2} \,.
\end{aligned}$$

We also need to calculate $\mathbb{E}[(Z_i - \bar{\boldsymbol{Z}})^4]$, we focus on the even terms, as we know terms with odd powers involving Z_i or $\bar{\boldsymbol{Z}}$ will be 0 due to symmetry around zero (as we have seen above). We also note that the fourth moment of Laplace(b) is $24b^4$. We have

$$\begin{aligned}
\mathbb{E}[(Z_i - \bar{\boldsymbol{Z}})^4] &= \mathbb{E}(Z_i^4) - 4\mathbb{E}(Z_i^3\bar{\boldsymbol{Z}}) + 6\mathbb{E}(Z_i^2\bar{\boldsymbol{Z}}^2) - 4\mathbb{E}(Z_i\bar{\boldsymbol{Z}}^3) + \mathbb{E}(\bar{\boldsymbol{Z}}^4) \\
&= \mathbb{E}(Z_i^4) - 4\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}(Z_i^3 Z_j) + 6\frac{1}{n^2}\sum_{j,k=1}^{n}\mathbb{E}(Z_i^2 Z_j Z_k) \\
&\quad - 4\frac{1}{n^3}\sum_{j,k,l=1}^{n}\mathbb{E}(Z_i Z_j Z_k Z_l) + \frac{1}{n^4}\sum_{j,k,l,q=1}^{n}\mathbb{E}(Z_j Z_k Z_l Z_q) \\
&= \mathbb{E}(Z_i^4) - \frac{4}{n}\mathbb{E}(Z_i^4) + 6\frac{1}{n^2}\left\{\mathbb{E}[Z_i^4] + (n-1)\mathbb{E}[Z_i^2]\mathbb{E}[Z_j^2]\right\} \\
&\quad - 4\frac{1}{n^3}\left\{\mathbb{E}[Z_i^4] + 3(n-1)\mathbb{E}[Z_i^2]\mathbb{E}[Z_k^2]\right\} + \frac{1}{n^4}\left\{n\mathbb{E}[Z_1^4] + 6n(n-1)\mathbb{E}[Z_j^2]\mathbb{E}[Z_l^2]\right\} \\
&= 24b^4\left(1 - \frac{4}{n} + \frac{6}{n^2} - \frac{3}{n^3}\right) + 24b^4\frac{n-1}{n^2}\left(1 - \frac{1}{n}\right) \\
&= 24b^4\left(1 - \frac{4}{n} + \frac{6}{n^2} - \frac{3}{n^3} + \frac{n-1}{n^2}\left(1 - \frac{1}{n}\right)\right) =: 24b^4 a_n
\end{aligned}$$

where $b = \dfrac{\Lambda}{\varepsilon}$. Note that $a_n > 0$ whenever $n > 1$. Thus

$$\begin{aligned}
&\mathbb{E}\left[S_f^2(\boldsymbol{x}, \boldsymbol{Z})\right] \\
&= \frac{1}{(n-1)^2}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^4 + \frac{1}{(n-1)^2}\sum_{i=1}^{n}\mathbb{E}(Z_i - \bar{\boldsymbol{Z}})^4 + \frac{6}{(n-1)^2}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^2\mathbb{E}(Z_i - \bar{\boldsymbol{Z}})^2 \\
&= \frac{1}{(n-1)^2}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^4 + \frac{n}{(n-1)^2}24b^4 a_n + \frac{6}{(n-1)^2}\frac{2b^2(n-1)}{n}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^2 \\
&= \frac{1}{(n-1)^2}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^4 + \frac{n}{(n-1)^2}24b^4 a_n + \frac{6}{n-1}\frac{2b^2}{n}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^2 .
\end{aligned}$$

Therefore

$$\begin{aligned}
\mathrm{MSE}(S_f(\boldsymbol{x}, \boldsymbol{Z})) &= \mathrm{Var}(S_f(\boldsymbol{x}, \boldsymbol{Z})) + (\mathbb{E}[S_f(\boldsymbol{x}, \boldsymbol{Z})])^2 = \mathbb{E}\left[S_f^2(\boldsymbol{x}, \boldsymbol{Z})\right] \\
&= \frac{1}{(n-1)^2}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^4 + \frac{n}{(n-1)^2}24b^4 a_n + \frac{6}{n-1}\frac{2b^2}{n}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^2 .
\end{aligned} \tag{5.25}$$

We now want to compare the variance of the post-processing and pre-processing mechanisms, and analyze the relationship between the privacy budgets. Let ε_0 be the privacy

budget for the output perturbation mechanism and ε be the privacy budget for the sanitized response mechanism. It should be recalled that the variance (and hence the MSE) for the output perturbation mechanism is

$$2\frac{\Lambda^4}{\varepsilon_0^2 n^2} .$$

We equate it to (5.25) and solve

$$\varepsilon = h(\varepsilon_0)$$

where h is some function. Of course, the function h depends on n, Λ and the data. We have

$$2\frac{\Lambda^4}{\varepsilon_0^2 n^2} = \frac{1}{(n-1)^2}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^4 + \frac{n}{(n-1)^2}24\frac{\Lambda^4}{\varepsilon^4}a_n + \frac{6}{n-1}\frac{2}{n}\frac{\Lambda^2}{\varepsilon^2}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^2 .$$

For simplicity, let $\varepsilon_0 = 1$, and $n > 1, \Lambda > 0$. Denote each term that does not involve ε as

$$A = \frac{2\Lambda^4}{n^2}, \quad B = \frac{1}{(n-1)^2}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^4, \quad C = \frac{n24\Lambda^4 a_n}{(n-1)^2}, \quad D = \frac{12}{n-1}\frac{\Lambda^2}{n\varepsilon^2}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^2 .$$

So we need to solve the following equation for ε:

$$\begin{aligned} A &= B + \frac{C}{\varepsilon^4} + \frac{D}{\varepsilon^2} \\ (A-B)\varepsilon^4 - D\varepsilon^2 - C &= 0 \end{aligned}$$

Since ε has to be positive, we obtain

$$\varepsilon = \sqrt{\frac{D + \sqrt{D^2 + 4(A-B)C}}{2(A-B)}} . \tag{5.26}$$

This equation provides the formal relationship between privacy for the pre- and post-processing. We notice that it is heavily dependent on the data.

To get some intuition, note that $\mathrm{MSE}(S_f(\boldsymbol{x}, \boldsymbol{Z}))$ is of the order

$$\frac{1}{n}\left(1 + \frac{1}{\varepsilon^2} + \frac{1}{\varepsilon}\right) ,$$

while $\mathrm{MSE}(O_f(\boldsymbol{x}, Z))$ is of the order $1/(n^2\varepsilon_0^2)$. Thus, to match the MSEs, we need to choose (set $\varepsilon_0 = 1$)) ε at the rate n. Hence, to keep the same data utility between pre- and post-processing, the post-processing has very little privacy (since ε has to be large).

Sample Median. As above, let $\boldsymbol{x} = (x_1, \cdots, x_n)$ be a database. We treat this database as fixed. Assume that the data come from a population with range $[0, \Lambda]$. We are interested in estimating the population median using the sample median.

Post-processing

Let $f(\boldsymbol{x}) = \text{median}(x_1, \ldots, x_n)$. The output perturbation mechanism is

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z \ ,$$

where Z is Laplace with variance

$$\text{Var}(Z) = 2\left(\frac{\Delta f}{\varepsilon}\right)^2 \ .$$

Recall from Example 2.1.3 that $\Delta f = \Lambda$.

Pre-processing

As above, in the case of pre-processing, we define $\boldsymbol{Y} = (x_1 + Z_1, \cdots, x_n + Z_n)$, where Z_i are independent with the distribution Laplace(Λ/ε). Then,

$$S_f(\boldsymbol{x}, \boldsymbol{Z})$$

is the sample median based on the observations $x_i + Z_i$, $i = 1, \ldots, n$.

In general the formulas for $\mathbb{E}[S_f(\boldsymbol{x}, \boldsymbol{Z})]$ and $\text{Var}(S_f(\boldsymbol{x}, \boldsymbol{Z}))$ are not feasible. Furthermore, the relationship between median$(\boldsymbol{x})$, median$(\boldsymbol{Z})$ and median$(\boldsymbol{Y})$ is not analytically tractable, except for a few cases when the data $\boldsymbol{x}$ are treated as random and come from a symmetric distribution.

To understand the noisy median estimator, we conduct some numerical experiments. We use a public dataset that contains values for age (amongst other variables, but we focus on age). We first randomize the variable age to date of birth (DOB), by adding a random value between 1 and 365 and dividing it by 365. We then aim to compare the median estimator for DOB for post-processing and pre-processing.

The privacy budget, ε, was set to 1. for both pre-processing and post-processing. In case of pre-processing, we add the Laplace noise to the database and calculate the median based on the noisy data. It is repeated $n = 1000$ times, producing 1000 medians. For post-processing, we calculate the median and then we add the Laplace noise. This procedure is repeated 1000 times.

For every query, the outputs for post-processing are centered around the true query value, which can be seen in Figure 5.15, denoted by the red line. However, as expected, the outputs for pre-processing are widely spread out, indicating much lower data utility when compared to the post-processing outputs. Similar to the mean, the median queries acting upon the privatized database are centered around the true parameter, indicating that the median private estimator is unbiased.

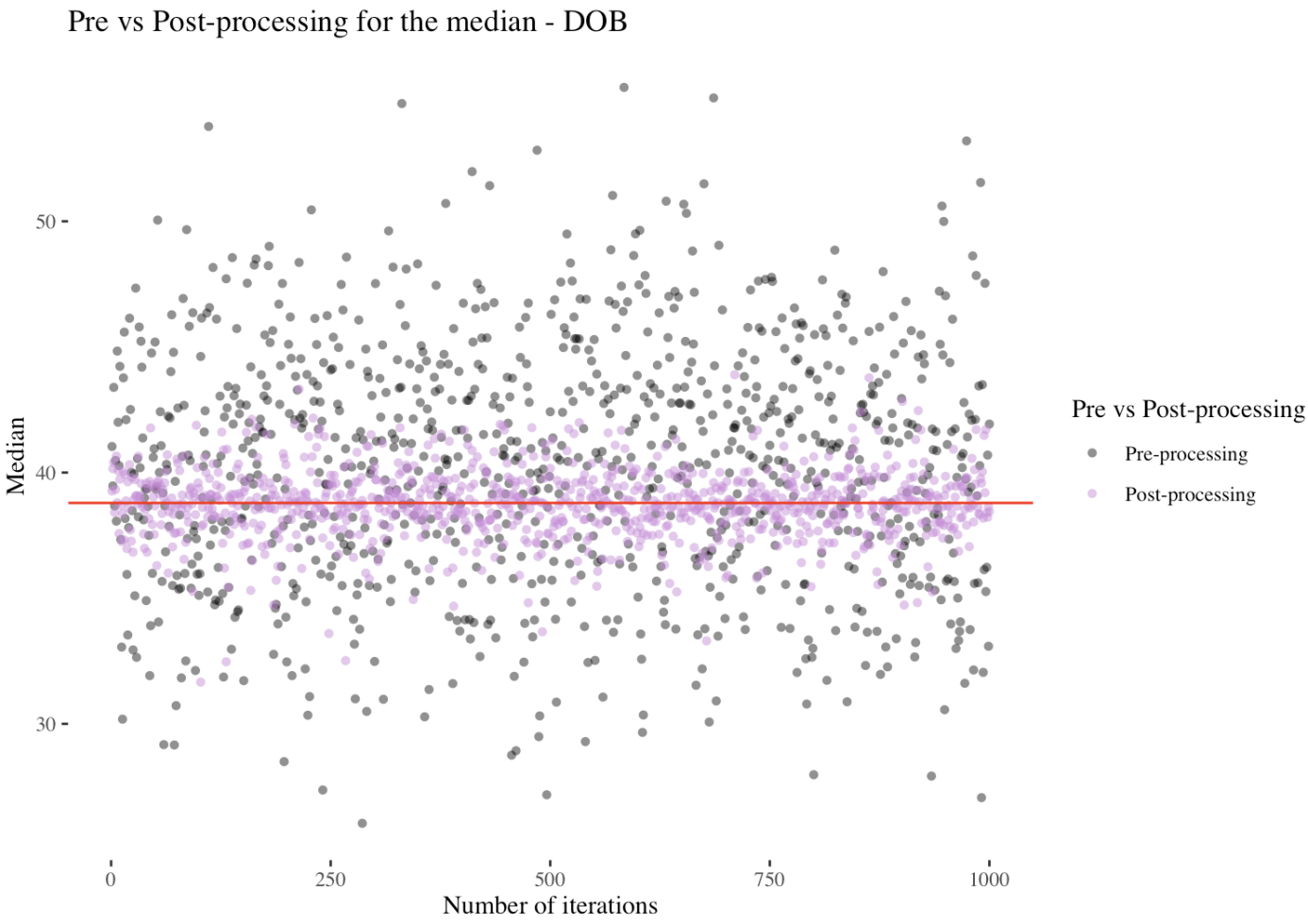

Figure 5.15: Post-processing vs Pre-processing for the median
The median estimator for pre-processing vs post-processing vs the true median for the variable date of birth.

Conclusion. It can therefore be concluded that the utility of post-processing vs. pre-processing is greater when both privacy budgets are fixed. This can be seen through the theoretical results and numerical experiments outlined in this section for the mean, median, and variance queries. Additionally, we observed through the theoretical calculations of the variance query that pre-processing induces a bias, which naturally lends to choosing post-processing when trying to maximize data utility. In conclusion, it is evident that when the scope of statistical analysis is constrained to a predefined set of statistical estimators, it is more advantageous from a practical standpoint to compute private statistics. Elastic sensitivity has been applied by Uber to differentially private queries, allowing the maximum data utility possible when computing data analytics over a set number of statistics, see [32]. Applying pre-processing to a database is only beneficial when one doesn't know who will query the database and what type of queries they

wish to compute.

5.7 Confidence Intervals

A natural progression from the previous Section 5.6 is to consider confidence intervals. In particular, we aim to develop private confidence intervals for various statistics, with a focus on the population mean μ. Despite the simplicity of the example, it already illustrates the challenges that arise in this context.

Assume that we have data $\boldsymbol{x} = (x_1, \ldots, x_n)$ coming from a population with mean μ and finite variance σ^2. In this context, $\boldsymbol{x}$ is considered as a random sample. It is assumed that the sample size n is sufficiently large. If the variance σ^2 is known, then the $(1-\alpha)$-confidence interval for the mean is give by

$$\left(\bar{\boldsymbol{X}} - z_{\alpha/2} \frac{\sigma}{\sqrt{n}}, \bar{\boldsymbol{X}} + z_{\alpha/2} \frac{\sigma}{\sqrt{n}} \right) .$$

We define the lower and upper bounds of the confidence interval as: $\bar{\boldsymbol{X}} - z_{\alpha/2} \frac{\sigma}{\sqrt{n}} =: CI_L$ and $\bar{\boldsymbol{X}} + z_{\alpha/2} \frac{\sigma}{\sqrt{n}} =: CL_R$. In terms of probability, we have:

$$\begin{aligned} &\mathbb{P}(CI_L \leq \mu \leq CI_U) = 1 - \alpha \, , \\ &\mathbb{P}\left(-z_{\alpha/2} \leq \frac{\bar{\boldsymbol{X}} - \mu}{\sigma/\sqrt{n}} \leq z_{\alpha/2} \right) = 1 - \alpha \, . \end{aligned}$$

In the event that the value of σ is unknown, it can be replaced with the standard deviation.

Confidence Interval for noisy mean. In the context of data privacy, we consider the sample mean query $f(\boldsymbol{x}) = \bar{\boldsymbol{x}}$. We observe $\bar{\boldsymbol{X}} + Z$, where Z is a random variable. In the event that the random variable Z is distributed according to the Laplace distribution, it is not feasible to obtain the distribution of the random variable $\bar{\boldsymbol{X}} + Z$. Indeed, let us assume that $\bar{\boldsymbol{X}}$ is normal with mean zero and variance σ^2/n. Then the convolution has a complicated density; see Example 4.5.9.

Therefore, in the present context, it is natural to consider a Gaussian Mechanism (see Theorem 4.4.3), whereby Z is assumed to have a centered normal distribution with variance σ_Z^2. In this case, $\bar{\boldsymbol{X}} + Z \sim \mathcal{N}(0, \frac{\sigma^2}{n} + \sigma_Z^2)$. We rewrite the expression for the

confidence interval with our noisy estimator:

$$\mathbb{P}\left(-z_{\alpha/2} \leq \frac{\bar{\boldsymbol{X}} + Z - \mu}{\sqrt{\frac{\sigma^2}{n} + \sigma_Z^2}} \leq z_{\alpha/2}\right) = 1 - \alpha \ ,$$

$$\mathbb{P}\left(-z_{\alpha/2}\sqrt{\frac{\sigma^2}{n} + \sigma_Z^2} \leq \bar{\boldsymbol{X}} + Z - \mu \leq z_{\alpha/2}\sqrt{\frac{\sigma^2}{n} + \sigma_Z^2}\right) = 1 - \alpha \ ,$$

$$\mathbb{P}\left(\bar{\boldsymbol{X}} + Z - z_{\alpha/2}\sqrt{\frac{\sigma^2}{n} + \sigma_Z^2} \leq \mu \leq \bar{\boldsymbol{X}} + Z + z_{\alpha/2}\sqrt{\frac{\sigma^2}{n} + \sigma_Z^2}\right) = 1 - \alpha \ .$$

The following observations can be made:

- The statistic $\bar{\boldsymbol{X}} + Z$ is observable, thereby enabling the computation of a confidence interval.
- According to Theorem 4.4.3, $\sigma_Z^2 \geq (c\Delta f/\varepsilon)^2$ with $c^2 > 2\ln(1.25/\delta)$. Thus, the information about σ_Z^2 is typically available to the user.

Using Theorem 4.4.3 we obtain immediately the following corollary.

Corollary 5.7.1. *Consider the output perturbation mechanism $\bar{\boldsymbol{X}} + Z$, where $Z \sim \mathcal{N}(0, \sigma_Z^2)$ with $\sigma_Z \geq c\Delta f/\varepsilon$, $c^2 > 2\ln(1.25/\delta)$. Assume that σ^2, the variance of the original population, is known. Then the $(1-\alpha)$-confidence interval*

$$\left(\bar{\boldsymbol{X}} + Z - z_{\alpha/2}\sqrt{\frac{\sigma^2}{n} + \sigma_Z^2}, \bar{\boldsymbol{X}} + Z + z_{\alpha/2}\sqrt{\frac{\sigma^2}{n} + \sigma_Z^2}\right)$$

is (ε, δ)-differentially private.

In practice, the value of σ^2 is unknown and *cannot* be inferred from the database, as the latter is not available. Instead, it may be the case that the user is in possession of the noisy sample variance. In other words, the user has access to

$$\widehat{\sigma}^2_{\text{noisy}} := S^2 + Z_1 := \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{\boldsymbol{X}})^2 + Z_1 = \widehat{\sigma}^2 + Z_1 \ ,$$

where Z_1 is a random variable. The statistics $\widehat{\sigma}^2$ and $\widehat{\sigma}^2_{\text{noisy}}$ are, respectively, non-noisy and noisy estimators of the population variance σ^2. Here, we may use a Laplace random variable, Z_1, with a parameter value of $\Delta f_1/\varepsilon_1$, where

$$f_1(\boldsymbol{x}) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})^2$$

is defined as the sample variance query. As previously stated in Example 5.5.2, it is possible that the resulting noisy sample variance may be negative. This issue will not be addressed further here. It is assumed that the database owner ensures that the estimator is strictly positive (for example, by sampling another copy of Z_1).

Given that two queries (sample mean, sample variance) are applied to the same database, we are in a position to utilize Lemma 4.5.13 to derive the following corollary.

Theorem 5.7.2. *Consider the output perturbation mechanism*

$$\left(\bar{\boldsymbol{X}} + Z, \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{\boldsymbol{X}})^2 + Z_1\right) =: (\bar{\boldsymbol{X}} + Z, \widehat{\sigma}^2 + Z_1) =: (\bar{\boldsymbol{X}} + Z, \widehat{\sigma}^2_{\text{noisy}}) \ ,$$

where

- $Z \sim \mathcal{N}(0, \sigma_Z^2)$ *with* $\sigma_Z \geq c\Delta f/\varepsilon$, $c^2 > 2\ln(1.25/\delta)$.
- Z_1 *is Laplace with the parameter* $\Delta f_1/\varepsilon_1$.

Then the confidence interval

$$\left(\bar{\boldsymbol{X}} + Z - z_{\alpha/2}\sqrt{\frac{\widehat{\sigma}^2_{\text{noisy}}}{n} + \sigma_Z^2}, \bar{\boldsymbol{X}} + Z + z_{\alpha/2}\sqrt{\frac{\widehat{\sigma}^2_{\text{noisy}}}{n} + \sigma_Z^2}\right)$$

is $(\varepsilon + \varepsilon_1, \delta)$*-differentially private.*

It should be noted that the confidence level was not indicated in the latter corollary. One might contend, however, that the confidence interval is asymptotically at the appropriate level, $1-\alpha$. Indeed, to illustrate this, we refer to Example 2.1.3 and Example 2.1.4. We have a population with the range $[0, \Lambda]$. Then $\Delta f = \Lambda/n$, while $\Delta f_1 = \Lambda^2/n$. Hence, we can write the confidence interval as

$$\left(\bar{\boldsymbol{X}} + Z - z_{\alpha/2}\sqrt{\frac{\widehat{\sigma}^2_{\text{noisy}}}{n} + \frac{c^2\Lambda}{\varepsilon n}}, \bar{\boldsymbol{X}} + Z + z_{\alpha/2}\sqrt{\frac{\widehat{\sigma}^2_{\text{noisy}}}{n} + \frac{c^2\Lambda}{\varepsilon n}}\right) \ .$$

Now, $\widehat{\sigma}^2_{\text{noisy}} = \widehat{\sigma}^2 + Z_1$. By the law of large numbers, $\widehat{\sigma}^2$ converges, as $n \to \infty$, to the population variance σ^2. On the other hand, since $\mathrm{E}[Z_1] = 0$ and $\mathrm{Var}(Z_1)$ is proportional to $(1/n^2)$,

$$Z_1 \xrightarrow{p} 0 \ ,$$

i.e., Z_1 converges to zero in probability. Hence, $\widehat{\sigma}^2_{\text{noisy}}$ converges in probability to the true variance, σ^2. Therefore, we conclude that the confidence interval is at the appropriate level.

To the contrary, if the sample variance is released based on pre-processing, then (5.23) indicates that the noisy sample variance does not converge to the population variance. Hence, the corresponding confidence interval is not at the level $1-\alpha$.

5.8 Changing the distance between probability distributions

Another relaxation of differential privacy, referred to as "concentrated differential privacy" (see [18]), was introduced in order to permit sharper analyses of several privacy-preserving computations. In [11] the authors proposed a version of this, based on the Rényi distance (cf. Definition 2.3.5). It is proposed as an intermediate notion between pure differential privacy and approximate differential privacy, with particular applicability to a normal noise. We will demonstrate that the latter does not seem to be the case.

We re-write the definition of zero concentrated privacy in the same language as Definition 4.2.1, where the bound in (5.27) corresponds to the bound (4.1). Recall that $\mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}}$ is the conditional distribution of the randomized mechanism $Q(\boldsymbol{X}, Z)$, given $\boldsymbol{X}=\boldsymbol{x}$.

Definition 5.8.1 (Zero-Concentrated Differential Privacy (zCDP).)**.** *Let $\boldsymbol{X}$ be a database, a random element of $\mathcal{D}$. Let Z be a random element with values in a metric space $\mathcal{E}$. Let $\xi, \rho > 0$. A randomized mechanism $Q : \mathcal{D} \times \mathcal{E} \to \mathbb{R}^d$ is zero-concentrated differentially private if $\forall\ \boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$, satisfying $d(\boldsymbol{x}, \boldsymbol{y}) = 1$, and $\alpha \in (1, \infty)$ we have*

$$D_\alpha(\mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}} \| \mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{y}}) \leq \xi + \rho\alpha\ , \tag{5.27}$$

where D_α is the α-Rényi divergence between the conditional distributions of Q given $\boldsymbol{x}$ and given $\boldsymbol{y}$.

Remark 5.8.2. We will write (ξ, ρ)-zCDP. In the case of $(0, \rho)$-zCDP, we will simply write ρ-zCDP.

Relation between zCDP, DP and Approximate DP. We present several results, both with and without accompanying proofs. These results demonstrate a relationship between the new version of differential privacy and the more classical ones.

Lemma 5.8.3 (ε-DP vs. zCDP)**.** *A mechanism $Q : \mathcal{D} \times \mathcal{E} \to \mathbb{R}^d$ satisfies ε-differential privacy if and only if it satisfies $(\varepsilon, 0)$-zCDP*

Proof. Let $\boldsymbol{x}, \boldsymbol{x'} \in \mathcal{D}$ be neighbouring datasets. Assume that Q satisfies ε-differential privacy. By monotonicity, and using (4.1), we can write:

$$D_\alpha(\mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}} \| \mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}'}) \leq D_\infty(\mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}} \| \mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}'}) \leq \varepsilon = \varepsilon + 0 \cdot \alpha \ ,$$

for all α. Therefore, Q satisfies $(\varepsilon, 0)$-zCDP. Conversely, suppose Q satisfies $(\varepsilon, 0)$-zCDP. Then we can write,

$$D_\infty(\mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}} \| \mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}'}) = \lim_{\alpha \to \infty} D_\alpha(\mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}} \| \mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}'}) \leq \lim_{\alpha \to \infty} \varepsilon + 0 \cdot \alpha = \varepsilon \ .$$

Thus, Q satisfies ε-differential privacy. □

We state the next result without a proof.

Lemma 5.8.4. *If Q satisfies ε-differential privacy, then Q satisfies $(0, \frac{1}{2}\varepsilon^2)$-zCDP.*

The subsequent result serves to establish the link between approximate differential privacy and zero-concentrated differential privacy.

Lemma 5.8.5 (Approximate DP vs. zCDP)**.** *Assume that $Q : \mathcal{D} \times \mathcal{E} \to \mathbb{R}^d$ satisfies (ξ, ρ)-zCDP. Then Q satisfies (ε, δ)-differential privacy for all $\delta > 0$ and*

$$\varepsilon = \xi + \rho + \sqrt{4\rho \log(1/\delta)} \ .$$

Thus to achieve a given (ε, δ)-differentially private guarantee it suffices to satisfy (ξ, ρ)-zCDP with

$$\rho = \left(\sqrt{\varepsilon - \xi + \log(1/\delta)} - \sqrt{\log(1/\delta)} \right)^2 \approx \frac{(\varepsilon - \xi)^2}{4 \log(1/\delta)} \ .$$

Proof. In what follows we will write $g(y \mid \boldsymbol{x})$ to denote the conditional density at point $y \in \mathbb{R}^d$ of $Q(\boldsymbol{X}, Z)$ given that $\boldsymbol{X} = \boldsymbol{x}$.

Let $\boldsymbol{x}, \boldsymbol{x'} \in \mathcal{D}$ be neighbouring databases. Define the function h as:

$$h(y) = \log \left(\frac{g(y \mid \boldsymbol{x})}{g(y \mid \boldsymbol{x'})} \right) .$$

Let $Y = Y(\boldsymbol{x})$ be a random variable with the distribution that is equal to the conditional distribution of $Q(\boldsymbol{X}, Z)$ given $\boldsymbol{X} = \boldsymbol{x}$ and let $U = h(Y)$. Fix $\alpha \in (1, \infty)$. Then,

$$\begin{aligned} \mathbb{E}\left[e^{(\alpha-1)U}\right] &= e^{(\alpha-1)D_\alpha(\mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}} \| \mathbb{P}_{Q|\boldsymbol{X}=\boldsymbol{x}'})} \\ &\leq e^{(\alpha-1)(\xi+\rho\alpha)} \ . \end{aligned}$$

By using Markov's inequality, we can express the probability as:

$$\begin{aligned} \mathbb{P}(U > \varepsilon) &= \mathbb{P}\left(e^{(\alpha-1)U} > e^{(\alpha-1)\varepsilon}\right) \\ &\leq \frac{\mathbb{E}[e^{(\alpha-1)U}]}{e^{(\alpha-1)\varepsilon}} \\ &\leq e^{(\alpha-1)(\xi+\rho\alpha-\varepsilon)} \ . \end{aligned}$$

Choose $\alpha = \dfrac{\varepsilon - \xi + \rho}{2\rho} > 1$, than the expression above becomes

$$\mathbb{P}(U > \varepsilon) \leq e^{-(\varepsilon-\xi-\rho)^2/4\rho} \leq \delta \, .$$

Then, for any measurable set $B \in \mathcal{B}(\mathbb{R}^d)$, we have

$$\begin{aligned}
&\mathbb{P}(Q(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{x}) = \mathbb{P}(Y \in B) \\
&\leq \mathbb{P}(\{Y \in B\} \cap \{U \leq \varepsilon\}) + \mathbb{P}(U > \varepsilon) \\
&\leq \int_{\mathbb{R}^d} g(y \mid \boldsymbol{x}) \cdot \mathbb{1}\{y \in B\} \cdot \mathbb{1}\{h(y) \leq \varepsilon\} dy + \delta \\
&\leq \int_{\mathbb{R}^d} e^{\varepsilon} g(y \mid \boldsymbol{x'}) \mathbb{1}\{y \in B\} dy + \delta \\
&= e^{\varepsilon} \mathbb{P}(Q(\boldsymbol{X}, Z) \in B \mid \boldsymbol{X} = \boldsymbol{x'}) + \delta \, .
\end{aligned}$$

Thus, we can conclude that (ε, δ)-differential privacy can be achieved via (ξ, ρ)-zCDP with $\rho \approx \frac{(\varepsilon-\xi)^2}{4\log(1/\delta)}$. □

Zero Concentrated DP for a Gaussian noise. The primary goal of introducing the revised definition of differential privacy was to facilitate the use of a gaussian noise mechanism. The following theorem corresponds to Theorem 4.4.3.

Theorem 5.8.6. *For any $f : \mathcal{D} \to \mathbb{R}$, the randomized output perturbation mechanism*

$$O_f(\boldsymbol{x}, Z) = f(\boldsymbol{x}) + Z$$

with the centered Gaussian noise with the variance σ^2 is $(0, (\Delta f)^2/2\sigma^2)$-zCDP.

Proof of Theorem 5.8.6. First, we evaluate the formula for the Rényi distance between normal laws with the same variance. We show that

$$D_\alpha \left(\mathcal{N}(\mu, \sigma^2) \| \mathcal{N}(\nu, \sigma^2) \right) = \frac{\alpha(\mu - \nu)^2}{2\sigma^2} \, .$$

We have

$$\begin{aligned}
&\exp\left((\alpha-1)D_\alpha\left(\mathcal{N}(\mu,\sigma^2)\|\mathcal{N}(\nu,\sigma^2)\right)\right)\\
&=\frac{1}{\sqrt{2\pi\sigma^2}}\int_{\mathbb{R}}\exp\left(-\alpha\frac{(x-\mu)^2}{2\sigma^2}-(1-\alpha)\frac{(x-\nu)^2}{2\sigma^2}\right)dx\\
&=\frac{1}{\sqrt{2\pi\sigma^2}}\int_{\mathbb{R}}\exp\left(-\frac{(x-(\alpha\mu+(1-\alpha)\nu))^2-(\alpha\mu+(1-\alpha)\nu)^2+\alpha\mu^2+(1-\alpha)\nu^2}{2\sigma^2}\right)dx\\
&=\left[\exp\left(\frac{(\alpha\mu-(1-\alpha)\nu)^2+\alpha\mu^2-(1-\alpha)\nu^2}{2\sigma^2}\right)\right]\underbrace{\frac{1}{\sqrt{2\pi\sigma^2}}\int_{\mathbb{R}}\exp\left(-\frac{(x-(\alpha\mu+(1-\alpha)\nu))^2}{2\sigma^2}\right)dx}\\
&=\exp\left(\frac{\alpha(\alpha-1)(\mu-\nu)^2}{2\sigma^2}\right).
\end{aligned}$$

The under braced expression is equal to 1.

Now, for neighbouring databases, the releases are normal, with means $f(\boldsymbol{x})$ and $f(\boldsymbol{y})$, which differ by at most Δf. The proof is now complete. □

The question thus arises *How useful is this proposed version of differential privacy?* The authors in [11] assert that this new approach improves the precision of the bounds, in particularly in the context of the normal algorithm. This claim will be disproved.

Remark 5.8.7. We consider the randomized algorithm with a normal noise with variance σ^2. Then, the result of Theorem 5.8.6 yields zCDP with the parameters $\xi = 0$ and $\rho = (\Delta f)^2/(2\sigma^2)$. Then, the result in Lemma 5.8.5 gives (ε_0, δ)-differential privacy with

$$\varepsilon_0 = \frac{(\Delta f)^2}{2\sigma^2} + \sqrt{2}\frac{(\Delta f)}{\sigma}\sqrt{\log(1/\delta)}\,.$$

We want to compare this obtained (ε_0, δ)-differential privacy to classical (ε, δ)-differential privacy. In other words, the aim is to ascertain which σ implies $\varepsilon_0 = \varepsilon$. This is a quadratic equation with a positive solution

$$\sigma^2 = \frac{(\Delta f)^2}{\varepsilon^2}\left(\sqrt{2}\sqrt{\log(1/\delta)} + \sqrt{2\log(1/\delta)+\varepsilon}\right)^2\,.$$

On the other hand, Theorem 4.4.3 yields (ε, δ)-differential privacy whenever

$$\sigma^2 > \sigma_0^2 = 2\ln(1.25/\delta)(\Delta f/\varepsilon)^2\,.$$

Thus, it is readily seen that application of the theory of zero-concentrated differential privacy results in a larger variance then the direct application of the classical theory in Theorem 4.4.3. As such, the claimed superiority of zCDP is doubtful.

5.9 Conclusion

In this chapter, we have explored the important balance between maintaining differential privacy guarantees and maximizing data utility. This line of research did not seem to be present in the literature (which focuses primarily on privacy aspects). While differential privacy provides a robust framework for privacy protection, it can be challenging to optimize data utility when injecting noise into a dataset or a query (statistic). Theoretically, this is an acceptable approach; however from a practical standpoint, data utility should be considered a primary objective. Our investigation focused on a number of mechanisms that satisfy differential privacy guarantees, including the traditional Laplace mechanism, Gaussian mechanism, and the novel Mixed Noise Mechanism (MNM), general sensitivity or blocking method.

We demonstrated through theoretical analysis and numerical experiments that the choice of noise mechanism may have a significant impact on the utility of the data, which in turn can affect the outcomes of real-world findings. The MNM, which dynamically selects between Laplace and Gaussian noise based on the sensitivity of the query, consistently showed superior performance in maintaining data utility compared to other mechanisms. This adaptive approach ensure that the noise added is minimized, while adhering to privacy constraints.

Our findings highlight the necessity of considering the specific statistical properties and sensitivity of the data when selecting differentially private mechanisms. By adapting the noise addition process to align with the intrinsic characteristics of the data, it is possible to achieve a more favourable dynamic between the protection of privacy and the utility of the data. This can be of particular importance in applications where the precision of the data is of high importance (for example, disease surveillance and fraud detection). Future work may involve further refinement of these mechanisms and exploring their applicability to a broader range of queries and data types.

Chapter 6

Time series

In this chapter we are interested in the privacy leakage associated with differentially private queries resulting from time series data. We focus on Vector AutoRegressive models. In terms of studying the effects of differential privacy in a time series setting, there is no an unified theory that we are aware of.

6.1 Introduction

In practical applications of differential privacy, it is essential to consider the temporal dependence of the underlying data (see e.g. [22] for differentially private traffic monitoring). However, to the best of our knowledge, major tech companies reset their privacy budgets when dealing with longitudinal observations, without accounting for temporal dependence. This deficiency in the current theoretical and practical framework may be attributed to the limited theoretical results available to measure the impact of dependence on the privacy budget in a differential privacy setting. For instance, [41] employs a Fourier Transform method, and [13] provides some theoretical bounds on privacy leakage. However, their methodology is not tied to specific time series models and also deal with discrete data, making its practical application unclear.

We focus on the privacy leakage associated with differentially private queries arising from time series data. The use of Vector Autoregressive (VAR) time series models allows for the demonstration of methods for adjusting the privacy budget in order to account for temporal dependence. The privacy budget will depend on the specific model parameters, which can be estimated using classical time series methodologies (see for example, [10]) with various existing software tools. In practice, once the data has been obtained, it is possible to fit a VAR model, estimate its parameters, and calculate the privacy budget using the formulas provided in this chapter.

Theoretically, the Gaussian mechanism is suitable in the time series context due to the linear structure of VAR models and the sum-closure property of normal distributions.

However, it would be challenging or even impossible to prove relevant results for the Laplace mechanism using our methodology. Indeed, while a sum of a finite number of dependent normal random variables remains normal, this sum-closure property does not hold for the Laplace distribution. In particular, the sum of two Laplace random variables is not longer Laplace. even worse, it violates differential privacy; see Example 4.5.7.

6.2 Differentially private queries in times series

Let T be a positive integer. Let

$$\boldsymbol{X}^{(t)} = (X_1^{(t)}, \cdots, X_n^{(t)})' \in \mathbb{R}^n \ , \quad t \in \{1, 2, \ldots, T\},$$

be an n-dimensional time series. Denote

$$\boldsymbol{X} = (\boldsymbol{X}^{(1)}, \cdots, \boldsymbol{X}^{(T)}) \in \mathbb{R}^{n \times T} \ .$$

Then, $\boldsymbol{X}$ is interpreted as a database of a group of n users. Each $\boldsymbol{X}^{(t)}$ is information about all the users at that particular time $t \in \{1, \ldots, T\}$, while $(X_i^{(1)}, \ldots, X_i^{(T)}) \in \mathbb{R}^T$ is information about the user $i \in \{1, \ldots, n\}$ for the entire period of time. For future reference, we also denote

$$\boldsymbol{X}_{(-1)}^{(t)} = (X_2^{(t)}, \cdots, X_n^{(t)})' \in \mathbb{R}^{n-1} \ .$$

In what follows, we will refer to $\boldsymbol{X}$ as either a *time series* or a *database*. There are two main situations to be considered:

- **Event-level privacy goal**. This is the situation in which the adversary is interested in the information about all individuals at a particular time point $t \in \{1, \ldots, T\}$. In discussing event-level privacy, it is assumed that the neighbouring databases are considered to be (without loss of generality)

  $$\boldsymbol{X}^{(t)} = (X_1^{(t)}, X_2^{(t)}, \cdots, X_n^{(t)})' \in \mathbb{R}^n$$

  and

  $$\boldsymbol{Y}^{(t)} = (Y_1^{(t)}, X_2^{(t)}, \cdots, X_n^{(t)})' = (Y_1^{(t)}, \boldsymbol{X}_{(-1)}^{(t)})' \in \mathbb{R}^n \ ,$$

  for a fixed $t \in \{1, 2, \ldots, T\}$.

- **User-level privacy goal**. This is the situation in which the adversary is interested in the data of a single individual at all time points $t \in \{1, \ldots, T\}$. When discussing user-level privacy, the neighbouring databases are considered to be

  $$\{X_1^{(1)}, \ldots, X_1^{(T)}\}$$

  and

  $$\{Y_1^{(1)}, \ldots, Y_1^{(T)}\} \ .$$

In what follows, we focus primarily on **event-level privacy**. It should be noted that if $T = 1$, the event-level privacy problem reduces to the classical one, where there is no time effect. The goal of this chapter is to study the effect of dependence on the privacy leakage.

6.2.1 Data release and attack scenarios

We fix $\ell \in \{1, \ldots, T\}$. We release a query at (some of) time points $1, \ldots, \ell$, and return a randomized response

$$(Q_1, \ldots, Q_\ell) := (Q_1(\boldsymbol{X}^{(1)}, Z^{(1)}), \ldots, Q_l(\boldsymbol{X}^{(l)}, Z^{(l)})) = (f_1(\boldsymbol{X}^{(1)}) + Z^{(1)}, \ldots, f_\ell(\boldsymbol{X}^{(\ell)}) + Z^{(\ell)}),$$

where $f_t : \mathbb{R}^n \to \mathbb{R}$ and $Z^{(t)}$, $t = 1, \ldots, T$, are independent random variables, independent from the time series $\boldsymbol{X}$. The random variables $Z^{(t)}$ will yield a level of privacy at the level ε_t when considered separately. We note in passing that in principle we should use the notation Q_{f_t} to indicate the dependence on the query f_t, but we write simply Q_t instead.

We note the following:

1. $Q_1, \ldots, Q_\ell$ are dependent, since they are functions of the time series $\boldsymbol{X}$;
2. $Q_1, \ldots, Q_\ell$ are *conditionally independent*, given the entire time series $\boldsymbol{X}$;
3. $Q_1, \ldots, Q_\ell$ are *not* conditionally independent, given a particular time stamp $\boldsymbol{X}^{(j)}$, for some $j \leq \ell$.

For the time series, we will assume the following **Vector AutoRegressive** relationship of order 1 (abbreviated as VAR(1)):

$$\boldsymbol{X}^{(t)} = \boldsymbol{A}\boldsymbol{X}^{(t-1)} + \boldsymbol{B}^{(t)}, \quad t = 1, \ldots, T\,, \tag{6.1}$$

where $\boldsymbol{X}^{(t)}$ are $(n \times 1)$ random vectors, $\boldsymbol{B}^{(t)}$ are random vectors of dimension $(n \times 1)$ and $\boldsymbol{A} = [a_{ij}]_{i,j=1}^n$ is a deterministic matrix of dimension $(n \times n)$. The VAR(1) model can be extended, in expense of more cumbersome notation and more involved calculations, to a general VAR(p) model. However, the linear structure of the VAR model is crucial. We note that we are not concerned with a stationarity of the model, hence there are no restrictions on the matrix $\boldsymbol{A}$.

Attack Scenarios

For the sake of simplicity, we will assume that $\ell = 2$. The methodology can be readily extended to include more time points, although this would involve complex notation and a more cumbersome computation. We consider several different attack scenarios:

A1. Adversary wants to learn about $x_i^{(1)}$, i.e. the value of the user i at time 1.

A2. Adversary wants to learn about $x_i^{(2)}$, i.e. the value of the user i at time 2.

A3. Adversary wants to learn about $\boldsymbol{x}^{(t)}$, i.e. the value of all records at time t, for $t = 1$ or $t = 2$.

We note that A1, A2, A3 fall into the category of *event-level privacy*.

Knowledge Scenarios

We also consider different adversary knowledge scenarios:

N1. The adversary knows Q_1 only (that is, the randomized query at time 1).

N2. The adversary knows Q_2 only (that is, the randomized query at time 2).

N3. The adversary knows both Q_1, Q_2.

We note that the combinations A1+N1 or A2+N2 result in classical differential privacy, and hence there will be omitted from further analysis.

6.3 Privacy leakage for time series

If $Z^{(t)}$, $t = 1, \ldots, \ell$, yield ε_t-privacy, the question is: what is the privacy of releasing $(Q_1, \ldots, Q_\ell)$? Intuitively, the worst case scenario (when observations at distinct time points are totally dependent) is $\varepsilon_1 + \cdots + \varepsilon_\ell$, and the best case scenario(when observations at distinct time points are independent) is $\max\{\varepsilon_1, \ldots, \varepsilon_\ell\}$. This is shown in the next two lemmas. For simplicity, we consider the case of $\ell = 2$ only.

Notation. For notational convenience, in what follows, $\mathbb{P}_Y(z)$ stands for the density of a continuous random variable Y evaluated at a point z. Likewise, $\mathbb{P}_{(Y_1,Y_2)}(z_1, z_2)$ represents the joint density of (Y_1, Y_2) at (z_1, z_2). Moreover, $\mathbb{P}_{Y_2|Y_1}(y_2 \mid y_1)$ is the conditional density of Y_2 at y_2 given $Y_1 = y_1$. This notation is slightly different as compared to the previous chapters. Sometimes, for notational convenience, we will write $(d/dz)\mathbb{P}(Y \leq z)$ for $\mathbb{P}_Y(z)$.

6.3.1 Total dependence and independence

Lemma 6.3.1 (Total Dependence)**.** *Let $a > 0$. Assume that the query f is linear. If $\boldsymbol{X}^{(2)} = a\boldsymbol{X}^{(1)}$ then releasing the randomized response (Q_1, Q_2) is $(\varepsilon_1 + \varepsilon_2/a)$-differentially private.*

Remark 6.3.2. The above lemma illustrates a potential drawback of differential privacy. If we multiply each entry in the dataset by $a > 1$, and then apply a query f to that dataset, we may need to add more noise in order to achieve the same level of privacy. This concept was previously discussed in DP Fallacy 5.1.4.

Proof of Lemma 6.3.1. If we have total dependence, we can write (Q_1, Q_2) as

$$(Q_1, Q_2) = (Q_1(\boldsymbol{X}^{(1)}, Z^{(1)}), Q_l(\boldsymbol{X}^{(2)}, Z^{(2)})) = (f(\boldsymbol{X}^{(1)}) + Z^{(1)}, f(a\boldsymbol{X}^{(1)}) + Z^{(2)}).$$

Then, due to the independence between random variables $Z^{(1)}$ and $Z^{(2)}$, we can calculate the conditional probabilities given $\boldsymbol{X}^{(1)} = \boldsymbol{x}^{(1)}$ and $\boldsymbol{X}^{(2)} = a\boldsymbol{x}^{(1)}$:

$$\frac{\mathbb{P}_{(Q_1(\boldsymbol{x}^{(1)},Z_1),Q_2(a\boldsymbol{x}^{(1)},Z_2))}(z_1, z_2)}{\mathbb{P}_{(Q_1(\boldsymbol{y}^{(1)},Z_1),Q_2(a\boldsymbol{y}^{(1)},Z_2))}(z_1, z_2)} = \frac{\mathbb{P}_{(f(\boldsymbol{x}^{(1)})+Z^{(1)})}(z_1)}{\mathbb{P}_{(f(\boldsymbol{y}^{(1)})+Z^{(1)})}(z_1)} \cdot \frac{\mathbb{P}_{(f(a\boldsymbol{x}^{(1)})+Z^{(2)})}(z_2)}{\mathbb{P}_{(f(a\boldsymbol{y}^{(1)})+Z^{(2)})}(z_2)} .$$

Using the definition of differential privacy, the first ratio is bounded by e^{ε_1}. Next, since f is linear, the function $g_a(\boldsymbol{x}) = f(a\boldsymbol{x})$ has the sensitivity $a\Delta f$. Thus, we can bound the second ratio by $e^{\varepsilon_2/a}$. □

Lemma 6.3.3 (Independence)**.** *If $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$ are independent with the same distribution, then releasing the randomized response (Q_1, Q_2) is $\max\{\varepsilon_1, \varepsilon_2\}$-differentially private.*

Proof. We first assume that a record has been removed from $\boldsymbol{X}^{(1)}$, while $\boldsymbol{X}^{(2)}$ is unchanged. Then, in the computation below, $\boldsymbol{y}^{(2)} = \boldsymbol{x}^{(2)}$. We can then write the ratio of probabilities as:

$$\frac{\mathbb{P}_{(Q_1(\boldsymbol{x}^{(1)},Z_1),Q_2(\boldsymbol{x}^{(2)},Z_2))}(z_1, z_2)}{\mathbb{P}_{(Q_1(\boldsymbol{y}^{(1)},Z_1),Q_2(\boldsymbol{y}^{(2)},Z_2))}(z_1, z_2)} = \frac{\mathbb{P}_{(f(\boldsymbol{x}^{(1)})+Z^{(1)})}(z_1)}{\mathbb{P}_{(f(\boldsymbol{y}^{(1)})+Z^{(1)})}(z_1)} \cdot \frac{\mathbb{P}_{(f(\boldsymbol{x}^{(2)})+Z^{(2)})}(z_2)}{\mathbb{P}_{(f(\boldsymbol{x}^{(2)})+Z^{(2)})}(z_2)} \leq e^{\varepsilon_1} .$$

A similar computation holds for the case when a record is removed from $\boldsymbol{X}^{(2)}$. □

6.3.2 Privacy leakage for the mean

This section examines the privacy leakage associated with differentially private queries. To gain insight into the effects, we have selected the mean query for our analysis. That is, for all $t \in \{1, \ldots, T\}$ and $\boldsymbol{x} = (x_1, \ldots, x_n)'$,

$$f(\boldsymbol{x}) \equiv f_t(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} x_i =: \overline{\boldsymbol{x}} .$$

We are going to assume that $t_1 = 1, \ldots, t_\ell = \ell$, $\ell \leq T$. Then, at any time $t = 1, \ldots, \ell$, we release the query

$$Q_t = \frac{X_1^{(t)} + \cdots + X_n^{(t)}}{n} + Z^{(t)} = \overline{\boldsymbol{X}}_n^{(t)} + Z^{(t)}.$$

Recall the matrix $\boldsymbol{A} = [a_{ij}]$ in the definition of the VAR(1) process. We introduce the notation

$$s_i = \sum_{j=1}^{n} a_{ji}, \quad i = 1, \dots, n. \tag{6.2}$$

Furthermore, we need to specify the parameters of the time series. Assume that

$$\boldsymbol{B}^{(2)} := (B_1^{(2)}, \dots, B_n^{(2)})' \sim \mathcal{N}(0, \Sigma_{\boldsymbol{B}}^{(2)}),$$

where $\Sigma_{\boldsymbol{B}}^{(2)} = (\sigma_{ij}^{(2)})_{i,j=1}^{n}$ is the variance-covariance matrix. Then

$$\overline{\boldsymbol{B}}_n^{(2)} := \frac{B_1^{(2)} + \cdots + B_n^{(2)}}{n} \sim \mathcal{N}\left(0, \sum_{i,j=1}^{n} \sigma_{ij}^{(2)}/n\right) =: \mathcal{N}(0, \mathrm{var}_{\boldsymbol{B}}^{(2)}/\mathrm{n}). \tag{6.3}$$

The noise added at times $t = 1, 2$ is assumed to be normally distributed, that is

$$Z^{(t)} \sim \mathcal{N}\left(0, 2\left(\frac{\Delta f}{\varepsilon_t}\right)^2 \ln\left(\frac{1.26}{\delta_t}\right)\right), \tag{6.4}$$

where Δf is a global sensitivity of the function f. We recall that this specification leads to $(\varepsilon_t, \delta_t)$-differential privacy, whenever a single query is returned. Indeed, by Theorem 4.4.3, the variance should strictly bigger than $2\left(\frac{\Delta f}{\varepsilon_t}\right)^2 \ln\left(\frac{1.25}{\delta_t}\right)$. Hence, 1.26 above is not a typo!

Next, for $x \in \mathbb{R}$ and $b_1, \dots, b_n$, we introduce the function:

$$g_{b_1,\dots,b_n}(x) = (b_1 + \cdots + b_n)x/n. \tag{6.5}$$

In what follows, we will consider different attack ("A") and knowledge ("N") scenarios introduced in Section 6.2. For each of the scenarios, we need to compare the appropriate conditional probabilities (see e.g. (6.10)-(6.11) below). The adversary knowledge leads to the consideration of specific outcomes (for example, we consider the outcomes $Q_1 = z_1, Q_2 = z_2$). In contrast, the learning goals A1-A3 lead to the consideration of particular forms of conditioning.

Scenario A1+N3

In this scenario, we know both Q_1, Q_2, the randomized query at times 1 and 2, respectively. We are interested in learning about $x_1^{(1)}$, which is the entry for user 1 at time 1. The main result of this section is Theorem 6.3.4.

Theorem 6.3.4. *Consider the time series model (6.1) with*

$$\boldsymbol{B} = (B_1^{(2)}, \ldots, B_n^{(2)})' \sim \mathcal{N}(0, \Sigma_{\boldsymbol{B}}^{(2)}).$$

Let $f(\boldsymbol{x}) = \frac{1}{n}\sum_{t=1}^{n} x_t$ *and assume that*

$$Z^{(t)} \sim \mathcal{N}\left(0, 2\left(\frac{\Delta f}{\varepsilon_t}\right)^2 \ln\left(\frac{1.26}{\delta_t}\right)\right).$$

For learning $x_1^{(1)}$*, the release*

$$(\overline{\boldsymbol{X}}_n^{(1)} + Z^{(1)}, \overline{\boldsymbol{X}}_n^{(2)} + Z^{(2)})$$

is $(\varepsilon_1 + \varepsilon_2', \delta_2')$*-DP with*

$$\varepsilon_2' = \varepsilon_2'(a_{11}, \ldots, a_{n1}) = \frac{\Delta g}{\sqrt{\left(\frac{\Delta f}{\varepsilon_2}\right)^2 + \frac{\mathrm{var}_{\boldsymbol{B}}^{(2)}}{2n\ln(1.26/\delta_2)}}} \tag{6.6}$$

and $\delta_2' = e^{\varepsilon_1}\delta_2 + e^{\varepsilon_2}\delta_1 + \delta_1\delta_2$*, where* Δg *is the sensitivity of*

$$g_{a_{11},\ldots,a_{n1}}(x) = \frac{(a_{11} + \cdots + a_{n1})}{n}x.$$

Proof. We start with $\boldsymbol{x}^{(1)} = (x_1^{(1)}, \ldots, x_n^{(1)})'$ and we want to learn about $x_1^{(1)}$. At $t = 2$ our information becomes

$$\boldsymbol{x}^{(2)} = \begin{pmatrix} a_{11}x_1^{(1)} + \cdots + a_{1n}x_n^{(1)} + B_1^{(2)} \\ \cdots \\ a_{n1}x_1^{(1)} + \cdots + a_{nn}x_n^{(1)} + B_n^{(2)} \end{pmatrix} = \boldsymbol{A}\boldsymbol{x}^{(1)} + \boldsymbol{B}^{(2)}. \tag{6.7}$$

We consider the idea of updating $x_1^{(1)}$, while keeping $\boldsymbol{x}_{(-1)}^{(1)} = (x_2^{(1)}, \ldots, x_n^{(1)})'$. Then, the corresponding information at the next time stamp also has to be updated. This becomes the user-level privacy problem. The neighbouring database becomes

$$\boldsymbol{y}^{(1)} = (y_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}) = (y_1^{(1)}, \boldsymbol{x}_{(-1)}^{(1)}). \tag{6.8}$$

That is, the information about all but the first user at time $t = 1$ remains unchanged.

Then

$$\boldsymbol{y}^{(2)} = \begin{pmatrix} a_{11}y_1^{(1)} + a_{12}x_2^{(1)} + \cdots + a_{1n}x_n^{(1)} + B_1^{(2)} \\ \cdots \\ a_{n1}y_1^{(1)} + a_{n2}x_2^{(1)} + \cdots + a_{nn}x_n^{(1)} + B_2^{(2)} \end{pmatrix}. \tag{6.9}$$

That is, at time $t = 2$, the information about *all* users change. The goal is to compare the following probabilities:

$$\mathbb{P}_{(Q_1,Q_2)|\boldsymbol{X}^{(1)}}(z_1, z_2 \mid \boldsymbol{x}^{(1)}) = \mathbb{P}_{(f(\boldsymbol{x}^{(1)})+Z^{(1)}, f(\boldsymbol{x}^{(2)})+Z^{(2)})}(z_1, z_2), \tag{6.10}$$

where $\boldsymbol{x}^{(2)}$ and $\boldsymbol{x}^{(1)}$ are related through (6.7), with

$$\mathbb{P}_{(Q_1,Q_2)|\boldsymbol{X}^{(1)}}(z_1, z_2 \mid \boldsymbol{y}^{(1)}) = \mathbb{P}_{(f(\boldsymbol{y}^{(1)})+Z^{(1)}, f(\boldsymbol{y}^{(2)})+Z^{(2)})}(z_1, z_2), \tag{6.11}$$

where $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(2)}$ are given in (6.8) and (6.9), respectively.

Since the random variables $Z^{(t)}$ are independent between themselves and also independent from the time series, as well as $(B_1^{(2)}, \ldots, B_n^{(2)})$ is independent of $(X_1^{(1)}, \ldots, X_n^{(1)})$, we can then write

$$\mathbb{P}_{(f(\boldsymbol{x}^{(1)})+Z^{(1)}, f(\boldsymbol{x}^{(2)})+Z^{(2)})}(z_1, z_2) = \mathbb{P}_{f(\boldsymbol{x}^{(1)})+Z^{(1)}}(z_1)\mathbb{P}_{f(\boldsymbol{x}^{(2)})+Z^{(2)}}(z_2)$$

and the latter expression is bounded by

$$(e^{\varepsilon_1}\mathbb{P}_{f(\boldsymbol{y}^{(1)})+Z^{(1)}}(z_1) + \delta_1)\mathbb{P}_{f(\boldsymbol{x}^{(2)})+Z^{(2)}}(z_2).$$

We need to bound the latter expression. Note that this cannot be bounded directly using the definition of differential privacy. We have

$$\begin{aligned} &\mathbb{P}_{f(\boldsymbol{x}^{(2)})+Z^{(2)}}(z_2) \\ &= \frac{d}{dz_2}\mathbb{P}\left(\frac{(\sum_{j=1}^n a_{j1})x_1^{(1)} + (\sum_{j=1}^n a_{j2})x_2^{(1)} + \cdots + (\sum_{j=1}^n a_{jn})x_n^{(1)}}{n} + \overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \le z_2\right) \\ &= \frac{d}{dz_2}\mathbb{P}\left(\frac{s_1x_1^{(1)} + s_2x_2^{(1)} + \cdots + s_nx_n^{(1)}}{n} + \overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \le z_2\right) \end{aligned}$$

with $s_1, \ldots, s_n$ given in (6.2). Set

$$z_2^* = z_2 - \frac{s_2x_2^{(1)} + \cdots + s_nx_n^{(1)}}{n}.$$

Then the last expression is

$$\frac{d}{dz}\mathbb{P}\left(\frac{(\sum_{j=1}^n a_{j1})x_1^{(1)}}{n} + \overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \le z_2^*\right).$$

Set

$$I = (e^{\varepsilon_1}\mathbb{P}_{f(\boldsymbol{y}^{(1)})+Z^{(1)}}(z_1) + \delta_1)\frac{d}{dz}\mathbb{P}\left(\frac{(\sum_{j=1}^n a_{j1})x_1^{(1)}}{n} + \overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \leq z_2^*\right).$$

Now, taking into account (6.3) and (6.4),

$$\overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \sim \mathcal{N}\left(0, 2\left(\frac{\Delta f}{\varepsilon_2}\right)^2 \ln\left(\frac{1.26}{\delta_2}\right) + \mathrm{var}_{\boldsymbol{B}}^{(2)}/\mathrm{n}\right).$$

Next, recall the definition (6.5) and set $g = g_{a_{11},\ldots,a_{n1}}$. Let $\Delta g := \Delta g_{a_{11},\ldots,a_{n1}}$ be the sensitivity of $g_{a_{11},\ldots,a_{n1}}$. Approximate differential privacy gives

$$\mathbb{P}_{g(x)+Z^*}(z) \leq e^{\varepsilon}\mathbb{P}_{g(y)+Z^*}(z) + \delta,$$

whenever $Z^* \sim \mathcal{N}\left(0, 2\left(\frac{\Delta g}{\varepsilon}\right)^2 \ln\left(\frac{1.26}{\delta}\right)\right)$. We fix $\delta = \delta_2$. Thus, the bound on I will follow from solving

$$2\left(\frac{\Delta g}{\varepsilon}\right)^2 \ln\left(\frac{1.26}{\delta_2}\right) = 2\left(\frac{\Delta f}{\varepsilon_2}\right)^2 \ln\left(\frac{1.26}{\delta_2}\right) + \mathrm{var}_{\boldsymbol{B}}^{(2)}/\mathrm{n}$$

with respect to ε. The solution, denoted by ε_2', is

$$\varepsilon_2' := \varepsilon_2'(a_{11},\ldots,a_{n1}) = \frac{\Delta g}{\sqrt{\left(\frac{\Delta f}{\varepsilon_2}\right)^2 + \frac{\mathrm{var}_{\boldsymbol{B}}^{(2)}}{2n\ln(1.26/\delta_2)}}}.$$

With this, we obtain

$$\begin{aligned}
&\frac{d}{dz}\mathbb{P}\left(\frac{(\sum_{j=1}^n a_{j1})x_1^{(1)}}{n} + \overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \leq z_2^*\right) \\
&\leq e^{\varepsilon_2'}\frac{d}{dz}\mathbb{P}\left(\frac{(\sum_{j=1}^n a_{j1})y_1^{(1)}}{n} + \overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \leq z_2^*\right) + \delta_2 = e^{\varepsilon_2'}\mathbb{P}_{f(\boldsymbol{y}^{(2)})+Z^{(2)}}(z_2) + \delta_2.
\end{aligned}$$

Hence,

$$\begin{aligned}
&\mathbb{P}_{(f(\boldsymbol{x}^{(1)})+Z^{(1)},f(\boldsymbol{x}^{(2)})+Z^{(2)})}(z_1,z_2) \\
&\leq (e^{\varepsilon_1}\mathbb{P}_{f(\boldsymbol{y}^{(1)})+Z^{(1)}}(z_1) + \delta_1)(e^{\varepsilon_2'}\mathbb{P}_{f(\boldsymbol{y}^{(2)})+Z^{(2)}}(z_2) + \delta_2) \\
&= e^{\varepsilon_1+\varepsilon_2'}\mathbb{P}_{(f(\boldsymbol{y}^{(1)})+Z^{(1)},f(\boldsymbol{y}^{(2)})+Z^{(2)})}(z_1,z_2) + (e^{\varepsilon_1}\delta_2 + e^{\varepsilon_2}\delta_1 + \delta_1\delta_2).
\end{aligned}$$

□

We make the following observations for consideration.

- The resulting privacy leakage, ε_2', is a function that depends on many model parameters.

- It can be demonstrated that if the parameters of $\Sigma_{\boldsymbol{B}}^{(2)}$ and $a_{21}, \ldots, a_{n1}$ are fixed, then ε_2' is a linear, increasing function of a_{11}. This is a very intuitive conclusion. As the value of a_{11} increases, the degree of dependence in the time series model also increases, resulting in a greater amount of information being leaked. In particular, it can be observed that the parameter ε_2' may be higher than ε_2 in certain instance3s. That is, in the VAR(1) time series model, adding normal noises with parameters ε_1 and ε_2 may lead to less privacy as compared to the situation of releasing the information subsequently. This is in the spirit of Remark 6.3.2. The dependence on the remaining model parameters is less obvious and will be studied below.

- If we assume that $a_{11} + \cdots + a_{n1} \leq 1$, we can expect that the resulting $\varepsilon_2' \leq \varepsilon_2$. This will be illustrated in the numerical analysis below.

Numerical analysis. The goal of this analysis is to study how the model parameters $a_{11}, \ldots, a_{n1}$ and $\sigma_{i,j}$ influence ε_2'. We consider the case when $n = 2$ and fix the following parameters.

- $\varepsilon_1 = \varepsilon_2 = 1$, $\delta_1 = \delta_2 = 0.05$;

- We will assume that $\Delta f = 1/2$ and $\Delta g = \dfrac{(a_{11} + a_{21})}{2}$. This corresponds to the possible range of the data being 1. (We note that this is not fully correct, since the data is generated using a normal law. However, it is sufficient for comparison purposes).

- We will set $\sigma_{1,1} = \sigma_{2,2} = 1$ and consider different values of $\sigma_{1,2} = \rho \in (-1, 1)$.

We conduct the following analysis:

- The parameters a_{11} and a_{21} are fixed at 1 and we consider ε_2' as a function of ρ. See Figure 6.1. We observe that $\varepsilon_2' < \varepsilon_2 = 1$ for any choice of ρ.

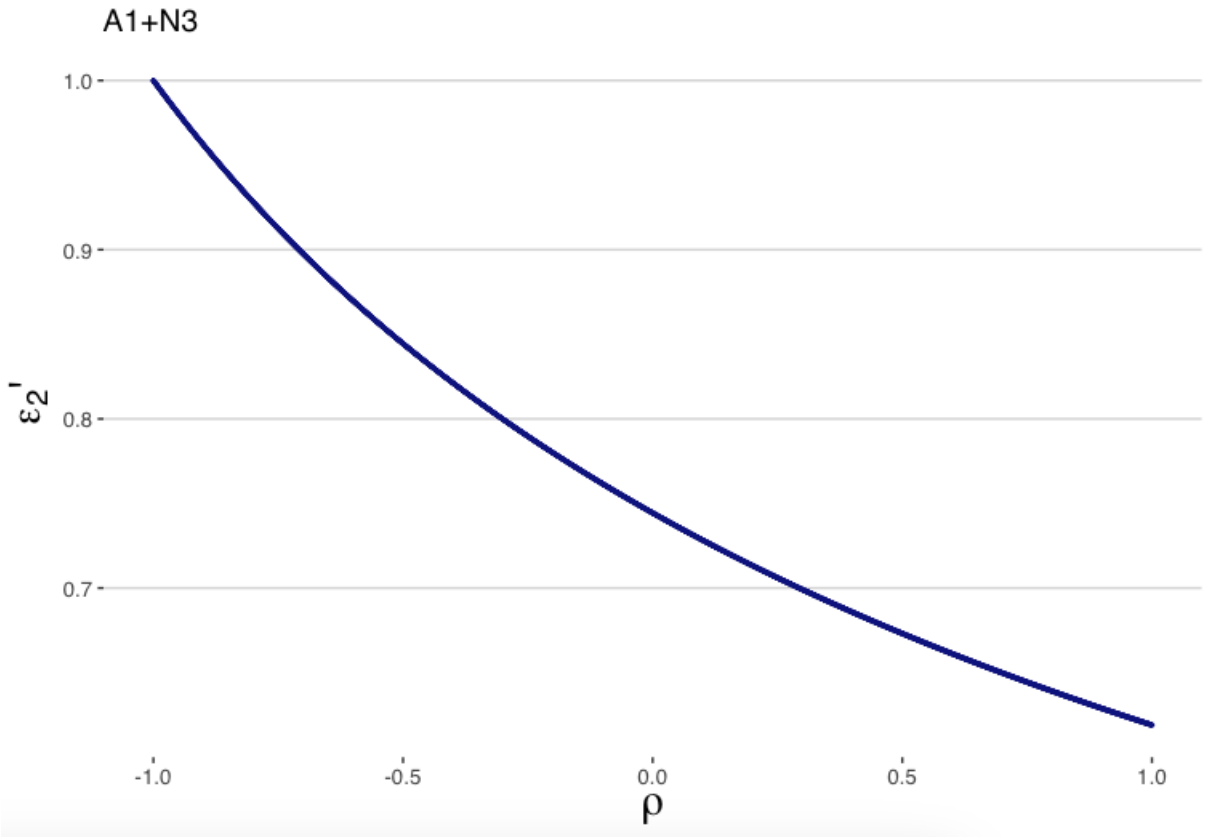

Figure 6.1: DP in time series: A1+N3 scenario
Relationship between ε_2' and ρ.

Scenario A1+N2

In this scenario, we know Q_2, and we want to learn about $x_1^{(1)}$.

Theorem 6.3.5. *Consider the time series model (6.1) with with*

$$(B_1^{(2)}, \ldots, B_n^{(2)})' \sim \mathcal{N}(0, \Sigma_{\boldsymbol{B}}^{(2)}).$$

Let $f(\boldsymbol{x}) = \frac{1}{n}\sum_{t=1}^{n} x_t$ *and assume that*

$$Z^{(t)} \sim \mathcal{N}\left(0, 2\left(\frac{\Delta f}{\varepsilon_t}\right)^2 \ln\left(\frac{1.26}{\delta_t}\right)\right).$$

For learning $x_1^{(1)}$*, the release*

$$\overline{\boldsymbol{X}}_n^{(2)} + Z^{(2)}$$

is $(\varepsilon_2', \delta_2)$*-DP with* ε_2' *given in (6.6).*

Proof. Following the proof of Theorem 6.3.4, the goal is to compare the following probabilities

$$\mathbb{P}_{Q_2|\boldsymbol{X}^{(1)}}(z_2 \mid \boldsymbol{x}^{(1)}) = \mathbb{P}_{(f(\boldsymbol{x}^{(2)})+Z^{(2)})}(z_2),$$

with

$$\mathbb{P}_{Q_2|\boldsymbol{X}^{(1)}}(z_2 \mid \boldsymbol{y}^{(1)}) = \mathbb{P}_{(f(\boldsymbol{y}^{(2)})+Z^{(2)})}(z_2),$$

where $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$, $\boldsymbol{y}^{(1)}$, $\boldsymbol{y}^{(2)}$ are the same as in the proof of Theorem 6.3.4. Hence, the result can be concluded immediately. □

Scenario A2+N1

In this scenario, we know Q_1, while we want to learn about $x_1^{(2)}$, the entry for the user 1 at time 2. In order to do this, we consider the idea of updating $x_1^{(2)}$ while keeping $x_{(-1)}^{(2)}$. Denote $\boldsymbol{y}^{(2)} = (y_1^{(2)}, x_{(-1)}^{(2)})$. Assume that the matrix $\boldsymbol{A}$ is invertible. Since $\boldsymbol{x}^{(2)} = \boldsymbol{A}\boldsymbol{x}^{(1)} + \boldsymbol{B}^{(2)}$, we get $\boldsymbol{x}^{(1)} = \boldsymbol{A}^{-1}\left(\boldsymbol{x}^{(2)} - \boldsymbol{B}^{(2)}\right)$. Likewise, $\boldsymbol{y}^{(1)} = \boldsymbol{A}^{-1}\left(\boldsymbol{y}^{(2)} - \boldsymbol{B}^{(2)}\right)$. Denote $\boldsymbol{A}^{-1} = [c_{ij}]_{i,j=1}^{n}$.

Theorem 6.3.6. *Consider the time series model (6.1) with*

$$(B_1^{(2)}, \ldots, B_n^{(2)})' \sim \mathcal{N}(0, \Sigma_{\boldsymbol{B}}^{(2)}).$$

Let $f(\boldsymbol{x}) = \frac{1}{n}\sum_{t=1}^{n} x_t$ *and assume that*

$$Z^{(t)} \sim \mathcal{N}\left(0, 2\left(\frac{\Delta f}{\varepsilon_t}\right)^2 \ln\left(\frac{1.26}{\delta_t}\right)\right).$$

For learning $x_1^{(2)}$*, the release*

$$\overline{\boldsymbol{X}}_n^{(1)} + Z^{(1)}$$

is $(\varepsilon_1', \delta_1)$*-DP with*

$$\varepsilon_1' = \frac{\Delta g}{\sqrt{\left(\frac{\Delta f}{\varepsilon_1}\right)^2 + \frac{\mathrm{var}_{\boldsymbol{r}'\boldsymbol{B}}^{(2)}}{2n\ln(1.26/\delta_1)}}},$$

where $\boldsymbol{r} = (r_1, \ldots, r_n)'$ *with* $r_i = \sum_{j=1}^{n} c_{ji}$*,* $i = 1, \ldots, n$ *and* Δg *is the sensitivity of*

$$g_{c_{11},\ldots,c_{n1}}(x) = \frac{(c_{11} + \cdots + c_{n1})}{n}x.$$

Proof. The goal is to compare the following probabilities

$$\mathbb{P}_{Q_1|\boldsymbol{X}^{(2)}}(z_1 \mid \boldsymbol{x}^{(2)}) = \mathbb{P}_{(f(\boldsymbol{x}^{(1)})+Z^{(1)})}(z_1),$$

with

$$\mathbb{P}_{Q_1|\boldsymbol{X}^{(2)}}(z_1 \mid \boldsymbol{y}^{(2)}) = \mathbb{P}_{(f(\boldsymbol{y}^{(1)})+Z^{(1)})}(z_1),$$

with $\boldsymbol{x}^{(1)} = \boldsymbol{A}^{-1}\left(\boldsymbol{x}^{(2)} - \boldsymbol{B}^{(2)}\right)$ and $\boldsymbol{y}^{(1)} = \boldsymbol{A}^{-1}\left(\boldsymbol{y}^{(2)} - \boldsymbol{B}^{(2)}\right)$. Then

$$\boldsymbol{x}^{(2)} = \begin{pmatrix} a_{11}x_1^{(1)} + \cdots + a_{1n}x_n^{(1)} + B_1^{(2)} \\ \cdots \\ a_{n1}x_1^{(1)} + \cdots + a_{nn}x_n^{(1)} + B_n^{(2)} \end{pmatrix} = \boldsymbol{A}\boldsymbol{x}^{(1)} + \boldsymbol{B}^{(2)},$$

$$\boldsymbol{y}^{(2)} = \begin{pmatrix} a_{11}y_1^{(1)} + \cdots + a_{1n}x_n^{(1)} + B_1^{(2)} \\ \cdots \\ a_{n1}y_1^{(1)} + \cdots + a_{nn}x_n^{(1)} + B_n^{(2)} \end{pmatrix},$$

$$\boldsymbol{x}^{(1)} = \boldsymbol{A}^{-1}\left(\boldsymbol{x}^{(2)} - \boldsymbol{B}^{(2)}\right) = \begin{pmatrix} c_{11}(x_1^{(2)} - B_1^{(2)}) + \cdots + c_{1n}(x_n^{(2)} - B_n^{(2)}) \\ \vdots \\ c_{n1}(x_1^{(2)} - B_1^{(2)}) + \cdots + c_{nn}(x_n^{(2)} - B_n^{(2)}) \end{pmatrix}$$

and

$$\boldsymbol{y}^{(1)} = \boldsymbol{A}^{-1}\left(\boldsymbol{y}^{(2)} - \boldsymbol{B}^{(2)}\right) = \begin{pmatrix} c_{11}(y_1^{(2)} - B_1^{(2)}) + \cdots + c_{1n}(x_n^{(2)} - B_n^{(2)}) \\ \vdots \\ c_{n1}(y_1^{(2)} - B_1^{(2)}) + \cdots + c_{nn}(x_n^{(2)} - B_n^{(2)}) \end{pmatrix}.$$

Let $r_i = \sum_{j=1}^{n} c_{ji}$, $i = 1, \ldots, n$, and $\mathbf{r} = (r_1, \ldots, r_n)$. Given that the random variables $Z^{(t)}$ are independent of one another and of the time series, and that $(B_1^{(2)}, \ldots, B_n^{(2)})$ is independent of $(X_1^{(1)}, \ldots, X_n^{(1)})$, it follows that

$$\begin{aligned} &\mathbb{P}_{(f(\boldsymbol{x}^{(1)})+Z^{(1)})}(z_1) \\ &= \frac{d}{dz_1}\mathbb{P}\left(\frac{r_1x_1^{(2)} + \cdots + r_nx_n^{(2)}}{n} - \frac{r_1B_1^{(2)} + \cdots + r_nB_n^{(2)}}{n} + Z^{(1)} \leq z_1\right) \\ &= \frac{d}{dz}\mathbb{P}\left(\frac{r_1x_1^{(2)}}{n} - \frac{r_1B_1^{(2)} + \cdots + r_nB_n^{(2)}}{n} + Z^{(1)} \leq z_1^*\right), \end{aligned}$$

where $z_1^* = z_1 - \frac{r_2 x_2^{(2)} + \cdots + r_n x_n^{(2)}}{n}$.
Since $\boldsymbol{B} = (B_1^{(2)}, \ldots, B_n^{(2)})'$ is multivariate normal with the mean vector zero and covariance matrix $\Sigma_{\boldsymbol{B}}^{(2)}$, we have

$$\mathbf{r}'\boldsymbol{B} = \sum_{i=1}^{n} r_i B_i^{(2)} \sim \mathcal{N}\left(0, \mathbf{r}^T \Sigma_{\boldsymbol{B}}^{(2)} \mathbf{r}\right) =: \mathcal{N}(0, \mathrm{var}_{\mathbf{r'B}}^{(2)}).$$

Now,

$$\frac{1}{n}\mathbf{r}'\boldsymbol{B} + Z^{(1)} \sim \mathcal{N}\left(0, 2\left(\frac{\Delta f}{\varepsilon_1}\right)^2 \ln\left(\frac{1.26}{\delta_1}\right) + \mathrm{var}_{\mathbf{r'B}}^{(2)}/\mathrm{n}\right).$$

Recall the definition (6.5) and set $g = g_{c_{11},\ldots,c_{n1}}$. We know that

$$\mathbb{P}_{(g(x)+Z^*)}(z) \leq e^{\varepsilon}\mathbb{P}_{(g(y)+Z^*)}(z) + \delta,$$

whenever $Z^* \sim \mathcal{N}\left(0, 2\left(\frac{\Delta g}{\varepsilon}\right)^2 \ln\left(\frac{1.26}{\delta}\right)\right)$. We fix $\delta = \delta_1$. Thus, the bound will follow from solving

$$2\left(\frac{\Delta g}{\varepsilon}\right)^2 \ln\left(\frac{1.26}{\delta_1}\right) = 2\left(\frac{\Delta f}{\varepsilon_1}\right)^2 \ln\left(\frac{1.26}{\delta_1}\right) + \mathrm{var}_{\mathbf{r'B}}^{(2)}/\mathrm{n}$$

with respect to ε. The solution, denoted by ε_1', is

$$\varepsilon_1' = \frac{\Delta g}{\sqrt{\left(\frac{\Delta f}{\varepsilon_1}\right)^2 + \frac{\mathrm{var}_{\mathbf{r'B}}^{(2)}}{2n\ln(1.26/\delta_1)}}}.$$

With this, we obtain

$$\begin{aligned}
&\frac{d}{dz}\mathbb{P}\left(\frac{(\sum_{j=1}^n c_{j1})x_1^{(2)}}{n} - \frac{\mathbf{r'B}}{n} + Z^{(1)} = z_1^*\right) \\
&\leq e^{\varepsilon_1'}\frac{d}{dz}\mathbb{P}\left(\frac{(\sum_{j=1}^n c_{j1})y_1^{(2)}}{n} - \frac{\mathbf{r'B}}{n} + Z^{(1)} = z_1^*\right) + \delta_1 \\
&= e^{\varepsilon_1'}\mathbb{P}_{(f(\boldsymbol{y}^{(2)})+Z^{(1)})}(z_1) + \delta_1.
\end{aligned}$$

□

Numerical analysis. The goal is to study how the model parameters $c_{ij}, i, j = 1, \ldots, n$ and $\sigma_{i,j}$ influence ε_1'. For this, we consider the case of $n = 2$ and fix the following parameters.

- $\varepsilon_1 = \varepsilon_2 = 1$, $\delta_1 = 0.05$;

- We will assume that $\Delta f = 1/2$ and $\Delta g = \dfrac{(c_{11} + c_{21})}{2}$. This corresponds to the possible range of the data being 1.
- We will set $\sigma_{1,1} = \sigma_{2,2} = 1$ and consider different values of $\sigma_{1,2} = \rho \in (-1, 1)$.

We conduct the following analysis:

- The parameters of matrix A are fixed in the following manner:

$$A = \begin{pmatrix} 0.5 & 0.1 \\ 0 & 0.5 \end{pmatrix},$$

  and we consider ε_1' as a function of ρ. See Figure 6.2 . It is obvious that $\varepsilon_1' < \varepsilon_1 = 1$, and approaches 0.7 as a minimum value. This is in line with the expectation that $\varepsilon_1' < \varepsilon_1$ and the relationship between the two parameters is non-linear. When $\rho = -1$, we observe that ε_1' and ε_1 are virtually the same.

- The parameters of matrix A are fixed in the following manner,

$$A = \begin{pmatrix} 0.9 & 0.5 \\ 0 & 0.9 \end{pmatrix},$$

  and we consider ε_1' as a function of ρ. See Figure 6.3. It is obvious that $\varepsilon_1' < \varepsilon_1 = 1$, and approaches 0.7 as a minimum value. This is in line with the expectation that $\varepsilon_1' < \varepsilon_1$ and the relationship between the two parameters is linear, unlike 6.2.

- We fix $\rho = 0$, or some other arbitrary value and play with the parameters of A in order to study the effects of the time series dependence structure. Define the matrix A as:

$$A = \begin{pmatrix} a_{11} & 0.5 \\ 0 & a_{22} \end{pmatrix},$$

  and consider ε_1' as a function of these coefficients. See Figures 6.4, 6.5. The results of varying the parameters a_{11} and a_{22} yield some unexpected and interesting outcomes. Two scenarios are considered fr the parameters a_{11}, a_{22} . The first scenario is defined by the conditions $a_{11} = a_{22} = \{0.1, \ldots, 0.9\}$. In other words, they are identical, and the values under consideration are arranged in ascending order. The outcomes were tested against three values of ρ, namely $\rho = -1, 0, 1$. As expected, When $\rho = -1$, we observe that $\varepsilon_1' < \varepsilon_1$. However, when $\rho = 0, 1$ we observe that for some values of $\Delta g = \dfrac{c_{11} + c_{21}}{2}$, $\varepsilon_1' > \varepsilon_1$. In fact, we observe local maxima, which suggest that there are optimal values of a_{11}, a_{22} that influence privacy leakage.

  In the second scenario we consider the case where a_{11}, a_{22} move in opposite directions. Specifically, we assume that a_{11} is the same as in the previous scenario,

while a_{22} takes on a range of values between $\{0.9, \ldots, 0.1\}$. This scenario further illustrates the unexpected behaviour that can arise when considering the impact of A on privacy leakage. As before, for some values of Δg, we have $\varepsilon_1' > \varepsilon_1$. Once again, we must consider the impact of matrix A on this outcome. Unlike the previous scenario, we do not observe local maxima, and thus must examine the behaviour of Δg in order to determine if a maximum exists.

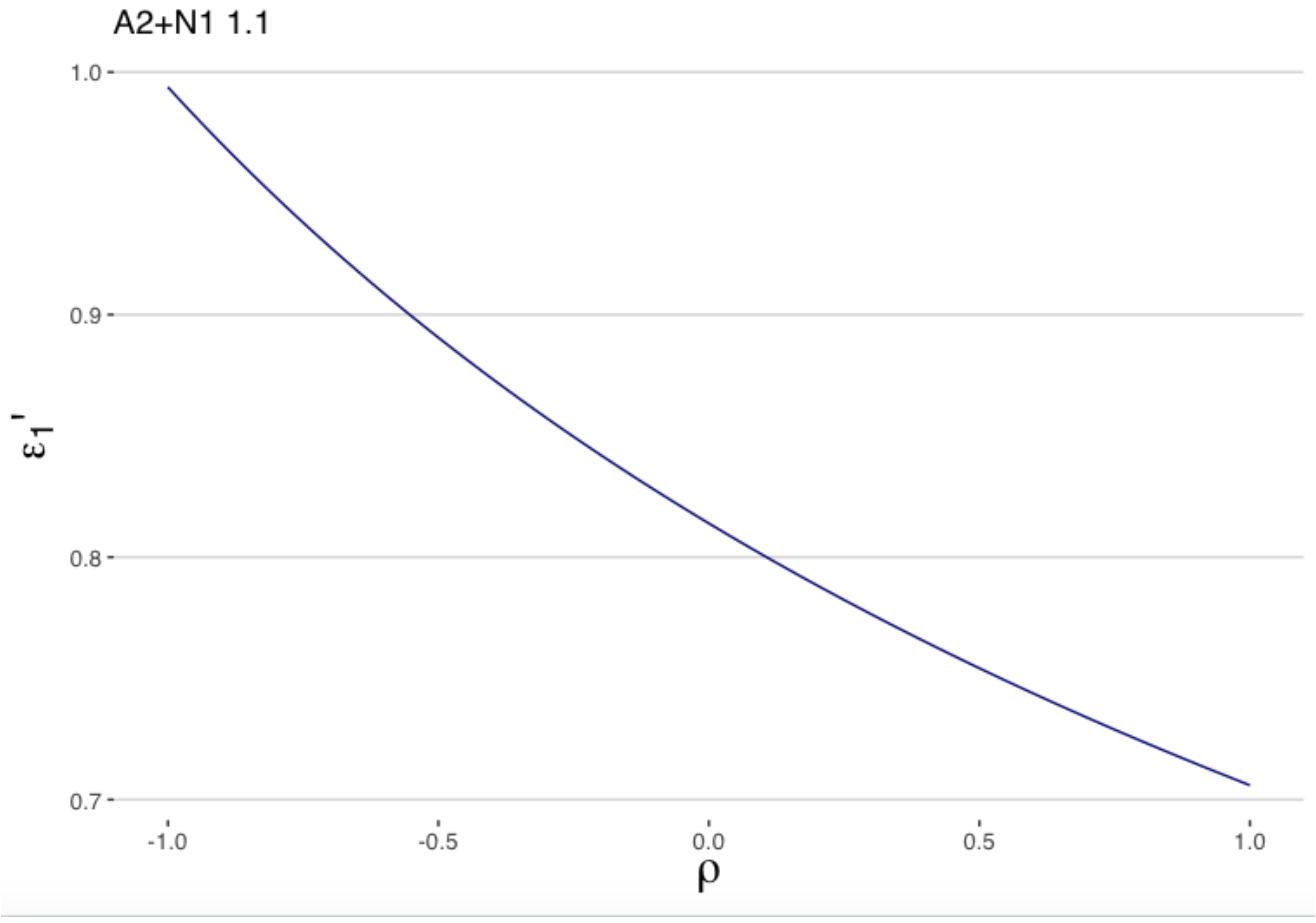

Figure 6.2: DP in time series: A2+N1 scenario; first example
Relationship between ε_1' and ρ.

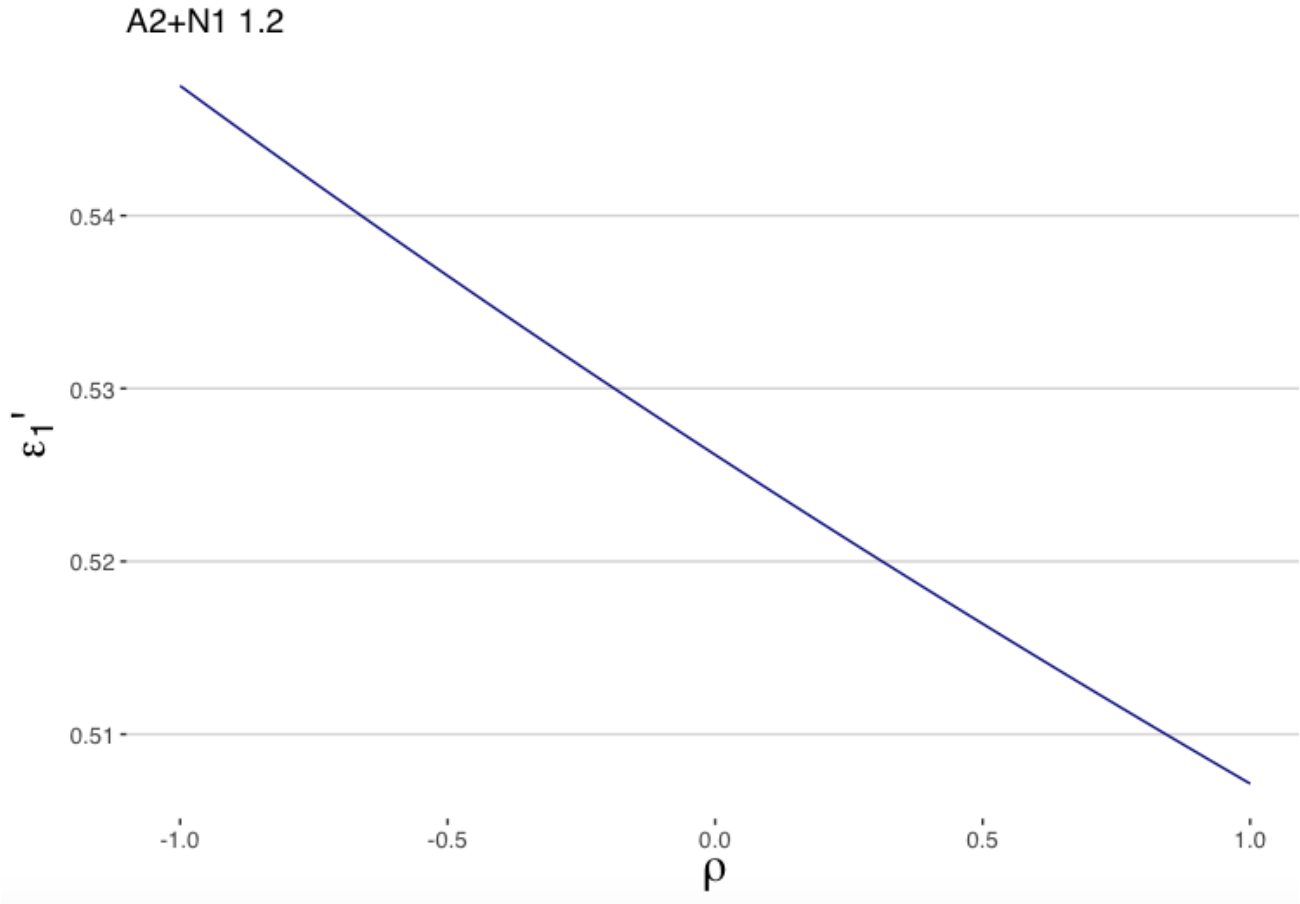

Figure 6.3: DP in time series: A2+N1 scenario; second example
Relationship between ε_1' and ρ.

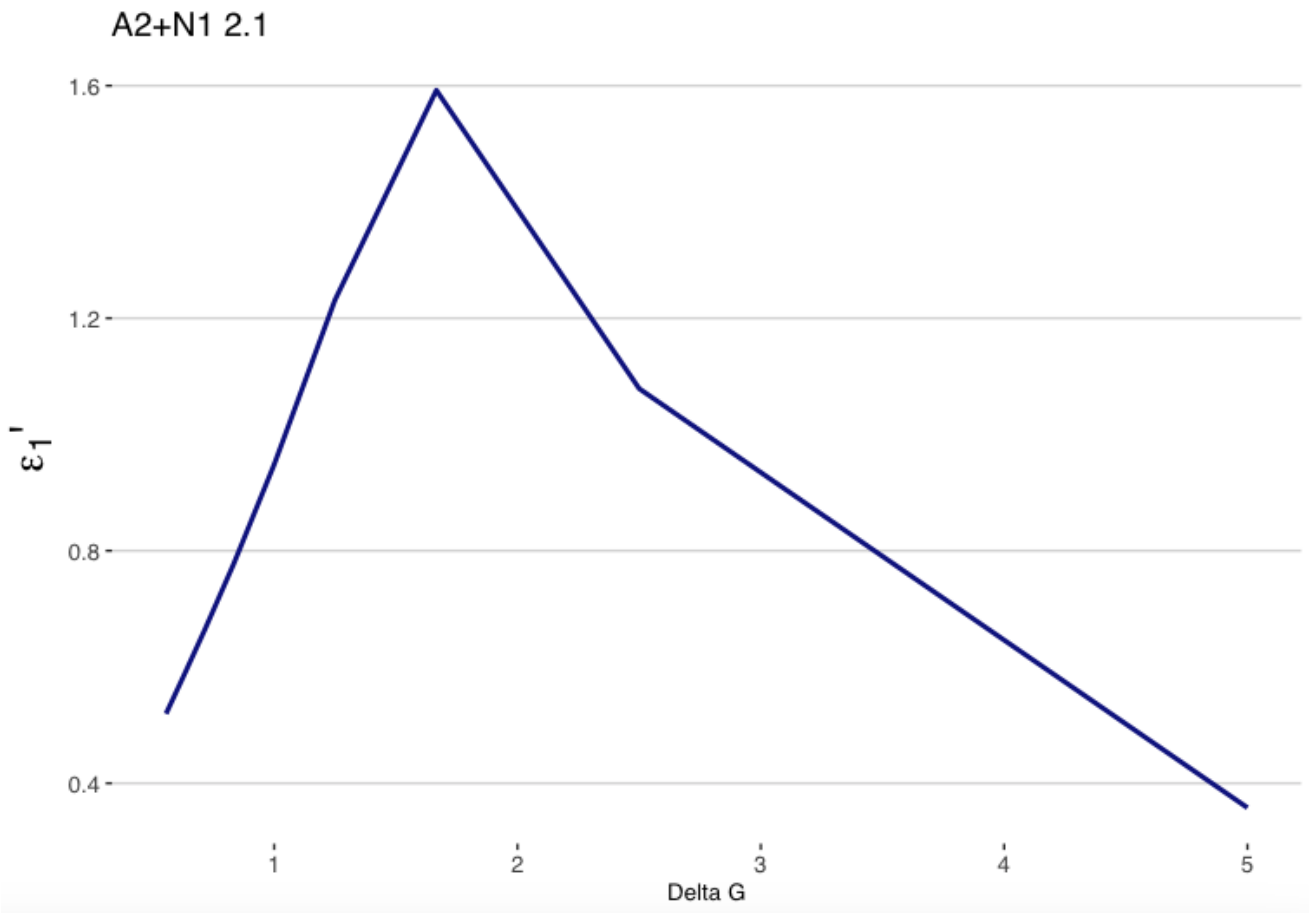

Figure 6.4: DP in time series: A2+N1 scenario; third example
Relationship between ε_1' and Δg, for different values of $a_{11} = a_{22} = \{0.1, \ldots, 0.9\}$ and for $\rho = -1, 0, 1$.

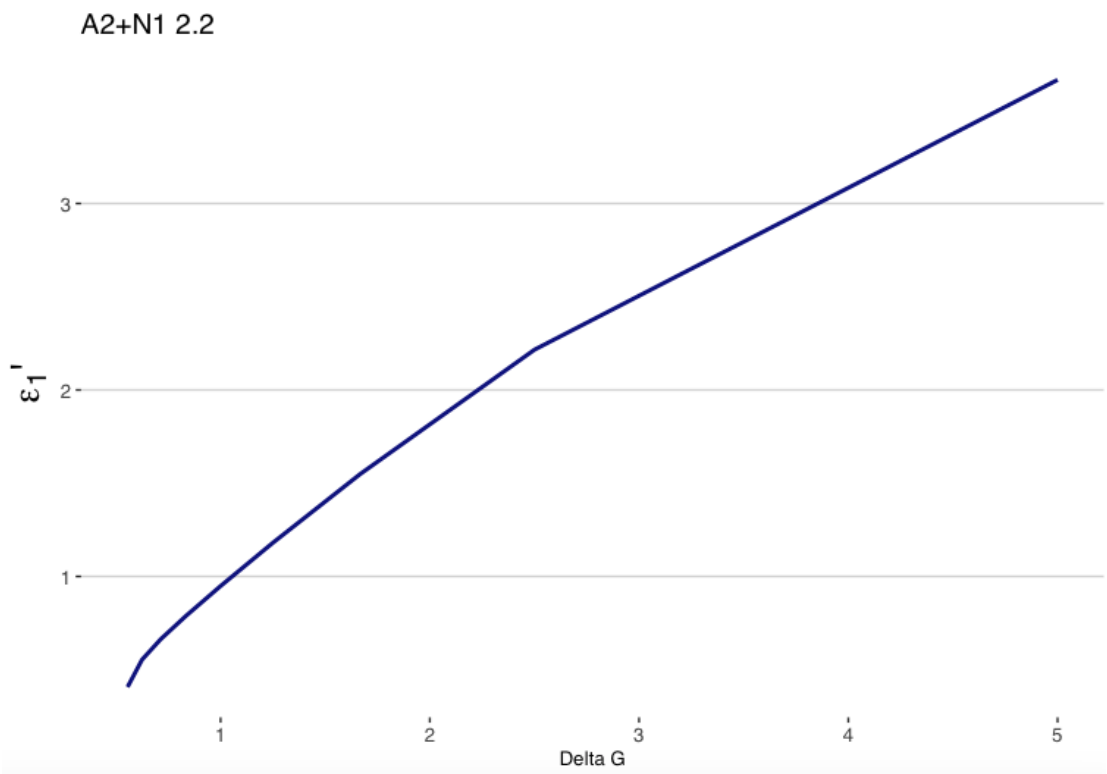

Figure 6.5: DP in time series: A2+N1 scenario; fourth example
Relationship between ε_1' and Δg, for different values of a_{11}, a_{22} and, for $\rho = -1, 0, 1$.

Scenario A4+N3

In this scenario, the adversary wishes to learn about $(x_1^{(1)}, x_1^{(2)})$, the values for the first user at all time points, and has the knowledge (Q_1, Q_2).

Theorem 6.3.7. *Consider the time series model (6.1) with with*

$$(B_1^{(2)}, \ldots, B_n^{(2)})' \sim \mathcal{N}(0, \Sigma_{\boldsymbol{B}}^{(2)}).$$

Let $f(\boldsymbol{x}) = \frac{1}{n}\sum_{t=1}^{n} x_t$ *and assume that*

$$Z^{(t)} \sim \mathcal{N}\left(0, 2\left(\frac{\Delta f}{\varepsilon_t}\right)^2 \ln\left(\frac{1.26}{\delta_t}\right)\right).$$

For learning $x_1^{(1)}, x_1^{(2)}$*, the release*

$$(\overline{\boldsymbol{X}}_n^{(1)} + Z^{(1)}, \overline{\boldsymbol{X}}_n^{(2)} + Z^{(2)})$$

is $(\varepsilon_1 + \varepsilon_2, \delta')$*-DP with* $\delta' = e^{\varepsilon_1}\delta_2 + e^{\varepsilon_2}\delta_1 + \delta_1\delta_2$.

Proof. The goal is to compare the following probabilities:

$$\mathbb{P}_{(Q_1,Q_2)|(\boldsymbol{X}^{(1)},\boldsymbol{X}^{(2)})}(z_1,z_2 \mid \boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)}) = \mathbb{P}_{(f(\boldsymbol{x}^{(1)})+Z^{(1)},f(\boldsymbol{x}^{(2)})+Z^{(2)})}(z_1,z_2),$$

with

$$\mathbb{P}_{(Q_1,Q_2)|(\boldsymbol{X}^{(1)},\boldsymbol{X}^{(2)})}(z_1,z_2 \mid \boldsymbol{y}^{(1)},\boldsymbol{y}^{(2)}) = \mathbb{P}_{(f(\boldsymbol{y}^{(1)})+Z^{(1)},f(\boldsymbol{y}^{(2)})+Z^{(2)})}(z_1,z_2),$$

Here, $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ are fixed, hence the time series structure is not relevant anymore. Therefore,

$$\begin{aligned}
&\mathbb{P}_{(Q_1,Q_2)|(\boldsymbol{X}^{(1)},\boldsymbol{X}^{(2)})}(z_1,z_2 \mid \boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)}) \\
&= \mathbb{P}_{f(\boldsymbol{x}^{(1)})+Z^{(1)}}(z_1)\mathbb{P}_{f(\boldsymbol{x}^{(2)})+Z^{(2)}}(z_2) \\
&\leq (e^{\varepsilon_1}\mathbb{P}_{f(\boldsymbol{y}^{(1)})+Z^{(1)}}(z_1)+\delta_1)(e^{\varepsilon_2}\mathbb{P}_{f(\boldsymbol{y}^{(2)})+Z^{(2)}}(z_2)+\delta_2) \\
&= e^{\varepsilon_1+\varepsilon_2}\mathbb{P}_{f(\boldsymbol{y}^{(1)})+Z^{(1)},f(\boldsymbol{y}^{(2)})+Z^{(2)}}(z_1,z_2) + (e^{\varepsilon_1}\delta_2 + e^{\varepsilon_2}\delta_1 + \delta_1\delta_2)\,.
\end{aligned}$$

□

Scenario A3+N3

In this scenario, we know both Q_1, Q_2, while we want to learn about $\boldsymbol{x}^{(t)}$, i.e. the value of all records at time t.

Theorem 6.3.8. *Consider the time series model (6.1) with with*

$$\boldsymbol{B} = (B_1^{(2)},\ldots,B_n^{(2)})' \sim \mathcal{N}(0,\Sigma_{\boldsymbol{B}}^{(2)}).$$

Let $f(\boldsymbol{x}) = \frac{1}{n}\sum_{t=1}^n x_t$ *and assume that*

$$Z^{(t)} \sim \mathcal{N}\left(0, 2\left(\frac{\Delta f}{\varepsilon_t}\right)^2 \ln\left(\frac{1.26}{\delta_t}\right)\right).$$

For learning $(x_1^{(1)},\ldots,x_n^{(1)})$*, the release*

$$(\overline{\boldsymbol{X}}_n^{(1)} + Z^{(1)}, \overline{\boldsymbol{X}}_n^{(2)} + Z^{(2)})$$

is $(\varepsilon_1+\varepsilon_2', \delta_2')$*-DP with*

$$\varepsilon_2' := \varepsilon_2'(a_{11},\ldots,a_{n1}) + \cdots + \varepsilon_2'(a_{1n},\ldots,a_{nn})$$

and δ_2*, where* $\varepsilon_2'(a_{1i},\ldots,a_{ni})$*,* $i=1,\ldots,n$*, are given in (6.6).*

Proof. We start with $\boldsymbol{x}^{(1)} = (x_1^{(1)}, \ldots, x_n^{(1)})'$ and we want to learn about $x_1^{(1)}, \cdots, x_n^{(1)}$. At $t = 2$ our information becomes as in (6.7). Now we consider updating all the values $x_1^{(1)}, \cdots, x_n^{(1)}$. Then, the corresponding information at the next time stamp also has to be updated. The "neighbouring" database becomes

$$\boldsymbol{y}^{(1)} = (y_1^{(1)}, y_2^{(1)}, \ldots, y_n^{(1)}). \tag{6.12}$$

We note that the neighbouring databases differ by n users. Then, the information at time $t = 2$ in the neighbouring database becomes

$$\boldsymbol{y}^{(2)} = \begin{pmatrix} a_{11}y_1^{(1)} + a_{12}y_2^{(1)} + \cdots + a_{1n}y_n^{(1)} + B_1^{(2)} \\ \cdots \\ a_{n1}y_1^{(1)} + a_{n2}y_2^{(1)} + \cdots + a_{nn}y_n^{(1)} + B_2^{(2)} \end{pmatrix} = \boldsymbol{A}\boldsymbol{y}^{(1)} + \boldsymbol{B}^{(2)}. \tag{6.13}$$

The goal is to compare the following probabilities:

$$\mathbb{P}_{(Q_1,Q_2)|\boldsymbol{X}^{(1)}}(z_1, z_2 \mid \boldsymbol{x}^{(1)}) = \mathbb{P}_{(f(\boldsymbol{x}^{(1)})+Z^{(1)}, f(\boldsymbol{x}^{(2)})+Z^{(2)})}(z_1, z_2),$$

where $\boldsymbol{x}^{(2)}$ and $\boldsymbol{x}^{(1)}$ are related through (6.7), with

$$\mathbb{P}_{(Q_1,Q_2)|\boldsymbol{X}^{(1)}}(z_1, z_2 \mid \boldsymbol{y}^{(1)}) = \mathbb{P}_{(f(\boldsymbol{y}^{(1)})+Z^{(1)}, f(\boldsymbol{y}^{(2)})+Z^{(2)})}(z_1, z_2),$$

where $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(2)}$ are given in (6.12) and (6.13), respectively.

The noise added at times $t = 1, 2$ is as in (6.4). We recall that in the context of the present theorem, this specification leads to $(n\varepsilon_t, \delta_t(n))$-differential privacy with

$$\delta_t(n) = \delta_t \sum_{j=0}^{n-1} \exp(j\varepsilon_t) \ ,$$

whenever a single query is used; see Lemma 4.5.12.

Since the random variables $Z^{(t)}$ are independent of one another and also independent from the time series, and that $(B_1^{(2)}, \ldots, B_n^{(2)})$ is independent of $(X_1^{(1)}, \ldots, X_n^{(1)})$, we can write

$$\begin{aligned}
&\mathbb{P}_{(f(\boldsymbol{x}^{(1)})+Z^{(1)}, f(\boldsymbol{x}^{(2)})+Z^{(2)})}(z_1, z_2) = \mathbb{P}_{f(\boldsymbol{x}^{(1)})+Z^{(1)}}(z_1)\mathbb{P}_{f(\boldsymbol{x}^{(2)})+Z^{(2)}}(z_2) \\
&\leq (e^{n\varepsilon_1}\mathbb{P}_{f(\boldsymbol{y}^{(1)})+Z^{(1)}}(z_1) + \delta_1(n)\mathbb{P}_{f(\boldsymbol{x}^{(2)})+Z^{(2)}}(z_2) \\
&= (e^{n\varepsilon_1}\mathbb{P}_{f(\boldsymbol{y}^{(1)})+Z^{(1)}}(z_1) + \delta_1(n)) \\
&\quad \times \frac{d}{dz_2}\mathbb{P}\left(\frac{s_1x_1^{(1)} + s_2x_2^{(1)} + \cdots + s_nx_n^{(1)}}{n} + \overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \leq z_2\right)
\end{aligned}$$

with $s_1, \ldots, s_n$ defined in (6.2).

The latter probability is compared to

$$\frac{d}{dz_2}\mathbb{P}\left(\frac{s_1 y_1^{(1)} + s_2 y_2^{(1)} + \cdots + s_n y_n^{(1)}}{n} + \overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \leq z_2\right).$$

In order to compare these probabilities we apply iteratively our results from Section 6.3.2. We have

$$\begin{aligned}
&\frac{d}{dz_2}\mathbb{P}\left(\frac{s_1 x_1^{(1)} + s_2 x_2^{(1)} + \cdots + s_n x_n^{(1)}}{n} + \overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \leq z_2\right) \\
&\leq e^{\varepsilon_2'(a_{11},\ldots,a_{n1})}\frac{d}{dz_2}\mathbb{P}\left(\frac{s_1 y_1^{(1)} + s_2 x_2^{(1)} + \cdots + s_n x_n^{(1)}}{n} + \overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \leq z_2\right) + \delta_2 \\
&\leq e^{\varepsilon_2'(a_{11},\ldots,a_{n1})+\varepsilon_2'(a_{12},\ldots,a_{n2})}\frac{d}{dz_2}\mathbb{P}\left(\frac{s_1 y_1^{(1)} + s_2 y_2^{(1)} + s_3 x_3^{(1)} + \cdots + s_n x_n^{(1)}}{n} + \overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \leq z_2\right) \\
&+ \delta_2(e^{\varepsilon_2'(a_{11},\ldots,a_{n1})} + 1) \\
&\leq e^{\varepsilon_2'}\frac{d}{dz_2}\mathbb{P}\left(\frac{s_1 y_1^{(1)} + s_2 y_2^{(1)} + s_3 y_3^{(1)} \cdots + s_n y_n^{(1)}}{n} + \overline{\boldsymbol{B}}_n^{(2)} + Z^{(2)} \leq z_2\right) + \delta_2(s_1, \ldots, s_n)
\end{aligned}$$

with

$$\varepsilon_2' := \varepsilon_2'(a_{11}, \ldots, a_{n1}) + \cdots + \varepsilon_2'(a_{1n}, \ldots, a_{nn}),$$

and

$$\delta_2(s_1, \cdots, s_n) := \delta_2\left(1 + e^{\varepsilon_2'(a_{11},\ldots,a_{n1})} + e^{\varepsilon_2'\sum_{j=1}^{2}(a_{1j},\ldots,a_{nj})} + \ldots + e^{\varepsilon_2'\sum_{j=1}^{n-1}(a_{1j},\ldots,a_{nj})}\right).$$

□

6.4 Conclusion

This chapter presents the development and exploration of theoretical frameworks and methodologies for understanding privacy leakage in differentially private queries derived from time series data. The necessity of adjusting the privacy budget to account for temporal dependence was demonstrated through the use of Vector Autoregressive (VAR) models, which also highlighted the impact of this dependence on the overall privacy guarantees.

The analysis demonstrated that temporal dependence in time series data presents considerable challenges to the maintenance of differential privacy. The Gaussian mechanism was identified as a particularly suitable approach in this context, given the linear

structure of VAR models and the closure properties of normal distributions. However, the difficulty of demonstrating the applicability of the Laplace mechanism in time series settings was also highlighted.

A comprehensive analysis was conducted to examine event-level and user-level privacy objectives, demonstrating how the extent of privacy leakage varies in different adversarial knowledge scenarios. The application of the proposed methodologies enables practitioners to estimate model parameters, fit appropriate time series models and calculate the adjusted privacy budget, thereby ensuring a balance between data utility and privacy protection.

The numerical analyses validated the theoretical results, demonstrating the influence of model parameters on privacy leakage. These findings underscore the importance of understanding the dependence structure in time series data when applying differential privacy mechanisms. The results indicate that careful consideration and adjustment of privacy budgets are crucial for effective privacy protection in practical applications involving time series data.

Chapter 7

Towards Machine Learning and Differential Privacy

7.1 Introduction

In the context of machine learning, differential privacy integration has become a popular form of mitigation for various privacy attacks. Techniques such as Differentially Private Stochastic Gradient Descent (DP-SGD) have been proposed to enhance the privacy of training models in deep learning and machine learning. DP-SGD, introduced in [1], modifies the standard SGD algorithm by adding noise to the gradients. However, some work on related problems for differential privacy in the context of empirical risk minimization has been done in [15].

Further extensions of DP-SGD have been proposed in [43], [4], [47]. Related work on Differentially Private Coordinate Descent algorithms (used in computing the LASSO or RIDGE solution) can be found in [36].

A summary of some recent work is presented in [5]. Recent research has focused on optimizing the privacy/utility trade-off using advanced techniques such as the smooth sensitivity framework, and on improving differentially private mechanisms for specific machine learning tasks. These areas have significant implications in fields such as healthcare, finance, and social networks; see e.g. [35].

When applying differential privacy in a machine learning setting, we face similar challenges to the results derived in Chapter 6 in the context of time series. In fact, Vector Autoregressive models studied in Chapter 6 and the Stochastic Gradient Descent have very similar Markov-type dynamics. Noise should be injected differently at each step, adaptively to the current model parameters (the learning rate in the context of DP-SGD). The aforementioned papers focus primarily on Gaussian noise and on constant level noise.

In addition, noise introduced for privacy reasons can degrade the accuracy of machine learning models. The aforementioned papers focus on the privacy aspects of DP-SGD algorithms, ignoring their utility. Fundamental questions such as the convergence of differentially private algorithms are not addressed.

As such, in this chapter we analyze with DP-SGD algorithm with adaptive noise added at each iteration. We provide privacy bounds (Theorem 7.6.1) and bounds on convergence of the algorithm (Theorem 7.7.1). Then, we analyze different scenarios when we can choose the privacy and the learning rate parameters in such the way that both privacy and convergence of the algorithm is guaranteed. To the best of our knowledge, it has not been studied in the literature yet.

As such, in this chapter we analyze with DP-SGD algorithm with adaptive noise added at each iteration. We provide privacy bounds (Theorem 7.6.1) and bounds on convergence of the algorithm (Theorem 7.7.1). Then, we analyze different scenarios when we can choose the privacy and the learning rate parameters in such the way that both privacy and convergence of the algorithm is guaranteed. To the best of our knowledge, it has not been studied in the literature yet.

The chapter is structured as follows. We start with preliminaries on (strongly) convex functions. In Section 7.3 we introduce the Gradient Descent algorithm, followed by its stochastic version in Section 7.5. The main results are included in Section 7.6 and Section 7.7. In the former, we provide privacy bounds on the Stochastic Gradient Algorithm. In the latter we prove convergence of the algorithm, using the classical techniques from [23]. In Section 7.8 we analyze our algorithm - we provide examples of adaptive learning rates that yield both privacy guarantees and convergence of the algorithm.

7.2 Preliminaries

Gradient and Stochastic Gradient Descent (SGD) are methods to find the minimum of a function of p variables. The set-up is as follows.

Let p be a positive integer. Let $\langle \cdot, \cdot \rangle$ be the inner product on $\mathbb{R}^p$ and $\| \cdot \|$ be a norm on $\mathbb{R}^p$. We denote $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p) \in \mathbb{R}^p$. If $p = 1$ we write θ instead of $\boldsymbol{\theta}$. Let $\psi : \mathbb{R}^p \to \mathbb{R}$ be a measurable function.

By $\boldsymbol{\theta}^*(\psi)$ we will denote the global (not necessarily unique) minimum of the function ψ, that is

$$\boldsymbol{\theta}^*(\psi) = \operatorname{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^p} \psi(\boldsymbol{\theta}). \tag{7.1}$$

If it is clear which function we consider, we will write $\boldsymbol{\theta}^*$ for $\boldsymbol{\theta}^*(\psi)$. Furthermore,

$$\psi^* := \psi(\boldsymbol{\theta}^*) \tag{7.2}$$

is the minimal value of the function. Since the minimum may not be unique, we will write $\operatorname{argmin}(\psi)$ to denote the set of all $\boldsymbol{\theta}$ that solve (7.1).

If $\psi : \mathbb{R}^p \to \mathbb{R}$ is differentiable, then $\nabla\psi$ is the gradient of the function ψ:

$$\nabla\psi(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial\psi}{\partial\theta_1}(\boldsymbol{\theta}) \\ \cdots \\ \frac{\partial\psi}{\partial\theta_p}(\boldsymbol{\theta}) \end{bmatrix} .$$

If $\psi : \mathbb{R}^p \to \mathbb{R}$ is twice differentiable, then $\nabla^2\psi$ is the Hessian of the function ψ:

$$\nabla^2\psi(\boldsymbol{\theta}) = \left[\frac{\partial^2\psi}{\partial\theta_i\partial\theta_j}\right]_{i,j=1}^{p} .$$

Convex functions. Convex and smooth functions play a special role in optimization.

Definition 7.2.1. *A function $\psi : \mathbb{R}^p \to \mathbb{R}$ is **convex** on a set $\mathcal{C}$ if for all $\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}} \in \mathcal{C}$ and $s \in [0,1]$ we have*

$$\psi(s\boldsymbol{\theta} + (1-s)\widetilde{\boldsymbol{\theta}}) \le s\psi(\boldsymbol{\theta}) + (1-s)\psi(\widetilde{\boldsymbol{\theta}}) .$$

Definition 7.2.2. *A differentiable function $\psi : \mathbb{R}^p \to \mathbb{R}$ is **convex** on $\mathbb{R}^p$ if for all $\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}} \in \mathbb{R}^p$ we have*

$$\widetilde{\boldsymbol{\theta}}^T \nabla^2\psi(\boldsymbol{\theta})\widetilde{\boldsymbol{\theta}} \ge 0 .$$

That is, the Hessian $\nabla^2\psi$ is positive semi-definite and hence all eigenvalues λ of the Hessian are nonnegative.

Properties of convex functions:

- If function ψ is differentiable and convex, then
$$\psi(\boldsymbol{\theta}) - \psi(\widetilde{\boldsymbol{\theta}}) \ge \langle \nabla\psi(\widetilde{\boldsymbol{\theta}}), \boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}} \rangle . \tag{7.3}$$
- If function ψ is differentiable and convex, then
$$(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T (\nabla\psi(\boldsymbol{\theta}) - \nabla\psi(\widetilde{\boldsymbol{\theta}})) \ge 0 .$$
- If ψ is differentiable and convex, then $\boldsymbol{\theta}^*$ is the global optimum if and only if
$$\langle \nabla\psi(\boldsymbol{\theta}^*), \boldsymbol{\theta} - \boldsymbol{\theta}^* \rangle \ge 0 . \tag{7.4}$$
Indeed, assume that (7.4) holds. Then, using convexity, we have
$$\psi(\boldsymbol{\theta}) \ge \psi(\boldsymbol{\theta}^*) + \langle \nabla\psi(\boldsymbol{\theta}^*), \boldsymbol{\theta} - \boldsymbol{\theta}^* \rangle \ge \psi(\boldsymbol{\theta}^*) .$$
- If the problem is unconstrained, e.g. $\mathcal{C} = \mathbb{R}^p$, then (7.4) reduces to $\nabla\psi(\boldsymbol{\theta}^*) = 0$.
- Convexity implies that local minima are also global minima.

Smooth functions.

Definition 7.2.3 (L-Lipschitz function)**.** *Let $\psi : \mathbb{R}^p \to \mathbb{R}$ and $L > 0$. We say that ψ is L-Lipschitz if or all $\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}} \in \mathbb{R}^p$,*

$$\|\psi(\boldsymbol{\theta}) - \psi(\widetilde{\boldsymbol{\theta}})\| \leq L\|\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}\| \ .$$

Definition 7.2.4 (L-smooth function)**.** *Let $\psi : \mathbb{R}^p \to \mathbb{R}$ and $L > 0$. We say that ψ is L-smooth if it is differentiable and for all $\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}} \in \mathbb{R}^p$,*

$$\|\nabla\psi(\boldsymbol{\theta}) - \nabla\psi(\widetilde{\boldsymbol{\theta}})\| \leq L\|\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}\| \ .$$

Sums of functions. Let $\psi_i : \mathbb{R}^p \to \mathbb{R}$ and define

$$\Psi(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \psi_i(\boldsymbol{\theta}) \ .$$

Obviously, if all ψ_i's are convex, then Ψ is convex as well. However, smoothness requires a little bit more care.

Lemma 7.2.5. *Assume that ψ_i, $i = 1, \ldots, n$, are L_i-smooth. Then Ψ is L_{ave}-smooth with $L_{\text{ave}} = \frac{1}{n}\sum_{i=1}^{n} L_i$.*

Convex, smooth functions.

Lemma 7.2.6. *If $\psi : \mathbb{R}^p \to \mathbb{R}$ is convex and L-smooth, then for all $\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}} \in \mathbb{R}^p$*

$$\frac{1}{L}\|\nabla\psi(\boldsymbol{\theta}) - \nabla\psi(\widetilde{\boldsymbol{\theta}})\|^2 \leq \langle \nabla\psi(\boldsymbol{\theta}) - \nabla\psi(\widetilde{\boldsymbol{\theta}}), \boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}} \rangle$$

and

$$\frac{1}{2L}\|\nabla\psi(\boldsymbol{\theta}) - \nabla\psi(\widetilde{\boldsymbol{\theta}})\|^2 \leq \psi(\boldsymbol{\theta}) - \psi(\widetilde{\boldsymbol{\theta}}) - \langle \nabla\psi(\widetilde{\boldsymbol{\theta}}), \boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}} \rangle \ .$$

7.3 Gradient Descent

Gradient Descent (GD) is an iterative procedure which is described in the following steps:

- Start with $\boldsymbol{\theta}_0$.
- For $t \geq 0$, $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \nabla\psi(\boldsymbol{\theta}_t)$, where $\eta_t > 0$ is the **learning rate**.
- Repeat N times until convergence is achieved.

The goal of GD is to descend towards the minimum,

$$\psi^* \leq \psi(\boldsymbol{\theta}_{t+1}) \leq \psi(\boldsymbol{\theta}_t) \ , \tag{7.5}$$

where ψ^* is the minimal value of the function ψ; see (7.2).
For convergence to be achieved, we can look at this from the perspective of theory and the perspective of the algorithm. Set $\delta > 0$, from the theoretical point of view we want to guarantee that

- $\psi(\boldsymbol{\theta}_N) - \psi^* \leq \delta$, or
- $\|\boldsymbol{\theta}_N - \boldsymbol{\theta}^*\| \leq \delta$, or
- $\min_{t=0,\ldots,N-1} \|\nabla\psi(\boldsymbol{\theta}_t)\|^2 < \delta$.

From the algorithm perspective, we stop the algorithm at step N whenever

- $\|\boldsymbol{\theta}_N - \boldsymbol{\theta}_{N-1}\| < \delta$, or
- $0 < \psi(\boldsymbol{\theta}_N) - \psi(\boldsymbol{\theta}_{N-1}) < \delta$, or
- $\|\nabla\psi(\boldsymbol{\theta}_N)\| < \delta$.

7.4 Stochastic optimization problem

Let $\psi : \mathbb{R}^p \to \mathbb{R}$ be a measurable function. Let x_i, $i = 1, \ldots, n$ be a sample from X, where X is a random vector in $\mathbb{R}^p$ with a distribution $F = F_X$. Set $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^{n\times p}$ and $\boldsymbol{X} = (X_1, \ldots, X_n) \in \mathbb{R}^{n\times p}$, where X_i have the same distribution as X. That is, in the thesis terminology, $\boldsymbol{x}$ is a database. We will treat database as fixed.

Functions ψ_i, $i = 1, \ldots, n$, will depend on data. Formally, we will et $\psi_i(\cdot) = \psi(\cdot; x_i)$. For $\boldsymbol{\theta} \in \mathbb{R}^p$ we define

$$\Psi(\boldsymbol{\theta}; \boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} \psi_i(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \psi(\boldsymbol{\theta}; x_i) \ .$$

We will usually drop the dependence on the data writing

$$\Psi(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \psi_i(\boldsymbol{\theta}) \ .$$

Then, $\Psi(\boldsymbol{\theta})$ is an empirical estimator of

$$\phi(\boldsymbol{\theta}) := \mathbb{E}[\Psi(\boldsymbol{\theta}; \boldsymbol{X})] = \mathbb{E}[\psi(\boldsymbol{\theta}; X)] \ . \tag{7.6}$$

Above, $\mathbb{E}$ is the expectation with respect to the distribution of X (recall that X_i come from the same distribution as X).

Define further

$$\widehat{\boldsymbol{\theta}}_n^* := \operatorname{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^p} \Psi(\boldsymbol{\theta}) \tag{7.7}$$

and

$$\widehat{\Psi}^* = \Psi(\widehat{\boldsymbol{\theta}}_n^*) \,. \tag{7.8}$$

Note that

- $\widehat{\boldsymbol{\theta}}_n^*$ depends on data $\boldsymbol{x} = (x_1, \ldots, x_n)$.
- $\widehat{\Psi}^*$ depends on data $\boldsymbol{x} = (x_1, \ldots, x_n)$.

In some elementary situations, the minimization problem can be solved explicitly by taking the gradient and solving $\nabla\Psi(\boldsymbol{\theta}) = 0$.

Example 7.4.1 (Sample mean)**.** Assume that X_i are random variables and have the same distribution as X. Here, the observations are $x_i \in \mathbb{R}$. Let $\boldsymbol{\theta} = \theta$ and

$$\phi(\boldsymbol{\theta}) = \mathbb{E}[(X - \theta)^2] \,.$$

Then

$$\psi_i(\boldsymbol{\theta}) = (x_i - \theta)^2$$

and

$$\Psi(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \theta)^2 \,.$$

Solving $\nabla\Psi(\boldsymbol{\theta}) = 0$ yields

$$\widehat{\boldsymbol{\theta}}_n^* = \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{\boldsymbol{x}} \,.$$

Example 7.4.2 (Linear regression and RIDGE)**.** Consider a linear regression problem $U_i = \theta_0 + \theta_1 V_i + \varepsilon_i$, where $X_i := (U_i, V_i)$ have the same distribution as $X = (U, V)$. Here, the observations are $x_i = (u_i, v_i) \in \mathbb{R}^2$. Let $\boldsymbol{\theta} = (\theta_0, \theta_1)$ and

$$\phi(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}[(U - \theta_0 - \theta_1 V)^2] + \frac{1}{2}\lambda\theta_1^2 \,.$$

Then

$$\psi_i(\boldsymbol{\theta}) = \frac{1}{2}(u_i - \theta_0 - \theta_1 v_i)^2 + \frac{\lambda}{2}\theta_1^2$$

and

$$\Psi(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}(u_i - \theta_0 - \theta_1 v_i)^2 + \frac{\lambda}{2}\theta_1^2 \, .$$

Hence,

$$\nabla\Psi(\boldsymbol{\theta}) = \left(-\frac{1}{n}\sum_{i=1}^{n}(u_i - \theta_0 - \theta_1 v_i), -\frac{1}{n}\sum_{i=1}^{n} v_i(u_i - \theta_0 - \theta_1 v_i) + \lambda\theta_1 \right)^T \, .$$

Solving $\nabla\Psi(\boldsymbol{\theta}) = 0$ yields $\widehat{\boldsymbol{\theta}}_n^* = (\widehat{\theta}_{0,n}^*, \widehat{\theta}_{1,n}^*)$ as the ridge estimator:

$$\widehat{\theta}_{1,n}^* = \frac{\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{\boldsymbol{v}})(u_i - \bar{\boldsymbol{u}})}{\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{\boldsymbol{v}})^2 + \lambda} \, , \quad \widehat{\theta}_{0,n}^* = \bar{\boldsymbol{u}} - \widehat{\theta}_{1,n}^* \bar{\boldsymbol{v}} \, . \tag{7.9}$$

7.5 Stochastic Gradient Descent (SGD)

We want to produce a sequence $\boldsymbol{\theta}_t$ that converges (in some sense) to $\widehat{\boldsymbol{\theta}}_n^*$. We could use $\nabla\Psi(\boldsymbol{\theta}_t)$ in each iteration, but it could be costly for large n and large p. Instead, at each time point t, we are going to approximate each $\Psi(\boldsymbol{\theta}_t)$ with $\widehat{\Psi}_t(\boldsymbol{\theta}_t)$ that will be defined below.

For each t, let I_t be random index sampled uniformly from $\{1, \ldots, n\}$, independently of everything else. At time t we will approximate $\nabla\Psi(\boldsymbol{\theta})$ by $\nabla\widehat{\Psi}_t(\boldsymbol{\theta})$ defined by

$$\nabla\widehat{\Psi}_t(\boldsymbol{\theta}) := \nabla\psi_{I_t}(\boldsymbol{\theta}) \, ,$$

Stochastic Gradient Descent:

- Start with $\boldsymbol{\theta}_0$.
- For $t \geq 0$, $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t)$, where η_t is the **learning rate** and

  $$\nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t) := \nabla\psi_{I_t}(\boldsymbol{\theta}_t) \, ,$$

  where I_t is a sequence of independent, identically distributed random variables sampled uniformly from $\{1, \ldots, n\}$.
- Repeat N times until convergence is achieved.

Note that the sequence $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_{t+1}(I_0, \ldots, I_t; \boldsymbol{x})$ depends on the random sequence $I_0, \ldots, I_t$ and the data $\boldsymbol{x}$ (that could be fixed or random). Let $\mathcal{F}_t = \sigma(I_0, \ldots, I_t)$, $t \geq 0$ and $\mathcal{X} = \sigma(\boldsymbol{x})$. Then $\boldsymbol{\theta}_{t+1}$ is $(\mathcal{X} \vee \mathcal{F}_t)$-measurable.

7.6 Differentially Private Stochastic Gradient Descent

We consider a version of the output perturbation mechanism. That is, we privatize the approximation $\boldsymbol{\theta}_t$ of $\boldsymbol{\theta}^*$. Instead of

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \nabla \psi_{I_t}(\boldsymbol{\theta}_t)$$

we consider

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \left(\nabla \psi_{I_t}(\boldsymbol{\theta}_t) + \gamma_t Z_t \right) \ , \tag{7.10}$$

where Z_t is a sequence of independent, identically distributed p-dimensional random vectors with the standard Laplace distribution and γ_t is a nonnegative sequence to be introduced below.

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two neighbouring databases. The functions $\psi_i(\boldsymbol{\theta})$ and hence the sequence $\boldsymbol{\theta}_t$ depend on the underlying database. We will need to make this dependence explicit. Furthermore, the sequence $\boldsymbol{\theta}_t$ is random. Hence we will write

$$\boldsymbol{\Theta}_{t+1}(\boldsymbol{x}) = \boldsymbol{\Theta}_t(\boldsymbol{x}) - \eta_t \left(\nabla \psi_{I_t}(\boldsymbol{\Theta}_t(\boldsymbol{x}); \boldsymbol{x}) + \gamma_t Z_t \right) \ ,$$

and

$$\boldsymbol{\Theta}_{t+1}(\boldsymbol{y}) = \boldsymbol{\Theta}_t(\boldsymbol{y}) - \eta_t \left\{ \nabla \psi_{I_t}(\boldsymbol{\Theta}_t(\boldsymbol{y}); \boldsymbol{y}) + \gamma_t Z_t \right\} \ .$$

In particular, $\psi_{I_t}(\boldsymbol{\theta}; \boldsymbol{x}) = \psi(\boldsymbol{\theta}, x_{I_t})$. This way we define the output-perturbation mechanism sequences

$$\mathcal{A}_t(\boldsymbol{x}) = \boldsymbol{\Theta}_t(\boldsymbol{x}) \ , \quad t = 1, 2, 3, \ldots ,$$

and

$$\mathcal{A}_t(\boldsymbol{y}) = \boldsymbol{\Theta}_t(\boldsymbol{y}) \ , \quad t = 1, 2, 3, \ldots .$$

The following theorem is the main result on DP-property of the Stochastic Gradient Descent.

Theorem 7.6.1. *Consider the privatized Stochastic Gradient Descent algorithm (7.10). Let $\Delta(\nabla\psi)$ be the global sensitivity of the gradient $\nabla\psi$. Assume that*

- *Z_t, $t = 1, 2 \ldots$, is the sequence of independent, identically distributed random vectors with the standard Laplace distribution;*
- *The constants γ_t are given by*

$$\gamma_1 = \Delta(\nabla\psi)/\varepsilon \ , \quad \gamma_t = \frac{1}{\beta_t} \left\{ \frac{\sum_{j=0}^{t-2} \eta_j}{\eta_{t-1}} + 1 \right\} \Delta(\nabla\psi)/\varepsilon \ , \quad t = 2, 3, \ldots \ ,$$

  where β_t is an arbitrary sequence of nonnegative numbers.

Then, the sequence $\mathcal{A}_t$, $t = 1, 2, \ldots$ is $\varepsilon\beta_t$-DP.

Proof of Theorem 7.6.1. Let $\boldsymbol{\theta}_0$ be fixed. We run the SGD algorithm for two neighbouring databases $\boldsymbol{x}$ and $\boldsymbol{y}$. We have $\boldsymbol{\Theta}_0(\boldsymbol{x}) = \boldsymbol{\Theta}_0(\boldsymbol{y}) = \boldsymbol{\theta}_0$. We have to keep in mind that we have two sources of randomness, the random index I_t and the private noise Z_t. Thus, we will consider conditioning $\mathbb{E}_{I_t}[\cdot] = \mathbb{E}[\cdot \mid I_t]$ on I_t. For the subsequent steps we need to condition not only on I_t, but also on the current value $\boldsymbol{\Theta}_t$. We then use the notation $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid I_t, \boldsymbol{\Theta}_t]$. In what follows, B is a Borel set in $\mathbb{R}^p$. Then, we write $cB + a$, $a \in \mathbb{R}^p$, $c \in \mathbb{R}$, for $cB + a = \{cb + a : b \in B\}$.

Step $t = 1$. At the first iteration $t = 1$ we have

$$\begin{aligned}\mathbb{P}(\mathcal{A}_1(\boldsymbol{x}) \in B) &= \mathbb{P}\left(\boldsymbol{\theta}_0 - \eta_1 \left(\nabla\psi_{I_1}(\boldsymbol{\theta}_0; \boldsymbol{x}) + \gamma_1 Z_1\right) \in B\right) \\ &= \mathbb{E}_{I_1}\left[\mathbb{P}\left(\gamma_1 Z_1 \in \frac{\boldsymbol{\theta}_0 - B}{\eta_0} - \nabla\psi_{I_1}(\boldsymbol{\theta}_0; \boldsymbol{x})\right)\right] \\ &= \mathbb{E}_{I_1}\left[\mathbb{P}\left(Z_1 \in \frac{\boldsymbol{\theta}_0 - B}{\gamma_1\eta_0} - \frac{1}{\gamma_1}\nabla\psi_{I_1}(\boldsymbol{\theta}_0; \boldsymbol{x})\right)\right] .\end{aligned}$$

Now, we are going to use the sliding property (5.5). Let

$$B_1 = \frac{\boldsymbol{\theta}_0 - B}{\gamma_1\eta_0} - \frac{1}{\gamma_1}\nabla\psi_{I_1}(\boldsymbol{\theta}_0; \boldsymbol{x})$$

and set

$$q = \frac{1}{\gamma_1}\left\{\nabla\psi_{I_1}(\boldsymbol{\theta}_0; \boldsymbol{x}) - \nabla\psi_{I_1}(\boldsymbol{\theta}_0; \boldsymbol{y})\right\} .$$

Recall that $\Delta(\nabla\psi)$ is the global sensitivity of the gradient $\nabla\psi$. We note that with the choice $\gamma_1 = \Delta(\nabla\psi)/\varepsilon$ we have

$$|q| = \frac{1}{\gamma_1}\left\{\nabla\psi(\boldsymbol{\theta}_0, x_{I_1}) - \nabla\psi(\boldsymbol{\theta}_0, y_{I_1})\right\} \leq \frac{1}{\gamma_1}\Delta(\nabla\psi) \leq \varepsilon .$$

Thus, using the sliding property,

$$\begin{aligned}&\mathbb{P}(\mathcal{A}_1(\boldsymbol{x}) \in B) = \mathbb{E}_{I_1}\left[\mathbb{P}\left(Z_1 \in B_1\right)\right] \\ &\leq e^{\varepsilon}\mathbb{E}_{I_1}\left[\mathbb{P}\left(Z_1 \in B_1 + q\right)\right] \\ &= e^{\varepsilon}\mathbb{E}_{I_1}\left[\mathbb{P}\left(Z_1 \in \frac{\boldsymbol{\theta}_0 - B}{\gamma_1\eta_0} - \frac{1}{\gamma_1}\nabla\psi_{I_1}(\boldsymbol{\theta}_0; \boldsymbol{x}) + \frac{1}{\gamma_1}\left\{\nabla\psi_{I_1}(\boldsymbol{\theta}_0; \boldsymbol{x}) - \nabla\psi_{I_1}(\boldsymbol{\theta}_0; \boldsymbol{y})\right\}\right)\right] \\ &= e^{\varepsilon}\mathbb{E}_{I_1}\left[\mathbb{P}\left(Z_1 \in \frac{\boldsymbol{\theta}_0 - B}{\gamma_1\eta_0} - \frac{1}{\gamma_1}\nabla\psi_{I_1}(\boldsymbol{\theta}_0; \boldsymbol{y})\right)\right] \\ &= e^{\varepsilon}\mathbb{P}(\mathcal{A}_1(\boldsymbol{y}) \in B) .\end{aligned}$$

In conclusion, the first iteration in the SGD algorithm is ε-DP.

Step $t = 2$. At the second iteration we have an additional term to control, since we do not necessary have $\boldsymbol{\Theta}_1(\boldsymbol{x}) \neq \boldsymbol{\Theta}_1(\boldsymbol{y})$. We have

$$\begin{aligned}\mathbb{P}(\mathcal{A}_2(\boldsymbol{x}) \in B) &= \mathbb{P}\left(\boldsymbol{\Theta}_1(\boldsymbol{x}) - \eta_1 \left(\nabla\psi_{I_2}(\boldsymbol{\Theta}_1(\boldsymbol{x}); \boldsymbol{x}) + \gamma_2 Z_2\right) \in B\right) \\ &= \mathbb{E}_2\left[\mathbb{P}\left(\gamma_2 Z_2 \in \frac{\boldsymbol{\Theta}_1(\boldsymbol{x}) - B}{\eta_1} - \nabla\psi_{I_2}(\boldsymbol{\Theta}_1(\boldsymbol{x}); \boldsymbol{x})\right)\right] \\ &= \mathbb{E}_2\left[\mathbb{P}\left(Z_2 \in \frac{\boldsymbol{\Theta}_1(\boldsymbol{x}) - B}{\gamma_2\eta_1} - \frac{1}{\gamma_2}\nabla\psi_{I_2}(\boldsymbol{\Theta}_1(\boldsymbol{x}); \boldsymbol{x})\right)\right] .\end{aligned}$$

Similarly to the first step, let

$$B_1 = \frac{\boldsymbol{\Theta}_1(\boldsymbol{x}) - B}{\gamma_2\eta_1} - \frac{1}{\gamma_2}\nabla\psi_{I_2}(\boldsymbol{\Theta}_1(\boldsymbol{x}); \boldsymbol{x})$$

and

$$q_2 := \frac{\boldsymbol{\Theta}_1(\boldsymbol{y}) - \boldsymbol{\Theta}_1(\boldsymbol{x})}{\gamma_2\eta_1} + \frac{1}{\gamma_2}\left\{\nabla\psi_{I_2}(\boldsymbol{\Theta}_1(\boldsymbol{x}); \boldsymbol{x}) - \nabla\psi_{I_2}(\boldsymbol{\Theta}_1(\boldsymbol{y}); \boldsymbol{y})\right\} .$$

We need to bound $|q_2|$. The second part is bounded by $\Delta(\nabla\psi)/\gamma_2$. For the first part we have

$$|\boldsymbol{\Theta}_1(\boldsymbol{y}) - \boldsymbol{\Theta}_1(\boldsymbol{x})| = \eta_0|\nabla\psi_{I_1}(\boldsymbol{\theta}_0, \boldsymbol{x}) - \nabla\psi_{I_1}(\boldsymbol{\theta}_0, \boldsymbol{y})| \leq \eta_0\Delta(\nabla\psi) .$$

Putting together,

$$|q_2| \leq \frac{1}{\gamma_2}\left\{\frac{\eta_0}{\eta_1} + 1\right\}\Delta(\nabla\psi) \leq \varepsilon\beta_2$$

with the choice

$$\gamma_2 = \frac{1}{\beta_2}\left\{\frac{\eta_0}{\eta_1} + 1\right\}\Delta(\nabla\psi)/\varepsilon .$$

We conclude by using the sliding property (5.5) in the same way as for step $t = 1$.

Step $t = n + 1$. Similarly to the proof of the step $t = 2$, we need to control

$$q_{n+1} := \frac{\boldsymbol{\Theta}_n(\boldsymbol{y}) - \boldsymbol{\Theta}_n(\boldsymbol{x})}{\gamma_{n+1}\eta_n} + \frac{1}{\gamma_{n+1}}\left\{\nabla\psi_{I_{n+1}}(\boldsymbol{\Theta}_n(\boldsymbol{x}); \boldsymbol{x}) - \nabla\psi_{I_{n+1}}(\boldsymbol{\Theta}_n(\boldsymbol{y}); \boldsymbol{y})\right\} .$$

As before, the second part is bounded by $\Delta(\nabla\psi)/\gamma_{n+1}$. For the first part, we bound it by induction. We claim that

$$|\boldsymbol{\Theta}_n(\boldsymbol{y}) - \boldsymbol{\Theta}_n(\boldsymbol{x})| \leq \left\{\sum_{j=0}^{n-1}\eta_j\right\}\Delta(\nabla\psi) . \tag{7.11}$$

We have already proved this for $n = 1$. The induction step is

$$\begin{aligned}&|\boldsymbol{\Theta}_n(\boldsymbol{y}) - \boldsymbol{\Theta}_n(\boldsymbol{x})| \\ &\leq |\left\{\boldsymbol{\Theta}_{n-1}(\boldsymbol{y}) - \boldsymbol{\Theta}_{n-1}(\boldsymbol{x})\right\}| + \eta_{n-1}|\left\{\nabla\psi_{I_n}(\boldsymbol{\Theta}_{n-1}(\boldsymbol{x}), \boldsymbol{x}) - \nabla\psi_{I_n}(\boldsymbol{\Theta}_{n-1}(\boldsymbol{y}), \boldsymbol{y})\right\}| \\ &\leq \left\{\sum_{j=0}^{n-2}\eta_j\right\}\Delta(\nabla\psi) + \eta_{n-1}\Delta(\nabla\psi) .\end{aligned}$$

Thus, (7.11) is proven. Therefore,

$$|q_{n+1}| \leq \frac{\left\{\sum_{j=0}^{n-1} \eta_j\right\}}{\gamma_{n+1}\eta_n}\Delta(\nabla\psi) + \frac{1}{\gamma_{n+1}}\Delta(\nabla\psi) \leq \varepsilon\beta_{n+1}$$

choosing

$$\gamma_{n+1} = \frac{1}{\beta_{n+1}}\left\{\frac{\sum_{j=0}^{n-1}\eta_j}{\eta_n} + 1\right\}\Delta(\nabla\psi)/\varepsilon\ .$$

□

7.7 Convergence of the algorithm

We have proved that the Stochastic Gradient Descent algorithm can be made differentially private. The next question is: can this DP-SGD algorithm converge? This is addressed in the next theorem. In what follows, $\|\cdot\|_2$ is the Euclidean norm on $\mathbb{R}^p$. Furthermore, we use several computations related to the conditional expectations. These properties can be found below in Section 7.9.

Theorem 7.7.1. *Assume that $\psi : \mathbb{R}^p \to \mathbb{R}$ is convex and L-smooth. Let Z_t be independent, identically distributed random vectors with the standard Laplace distribution on $\mathbb{R}^p$, independent of everything else. Let $\eta_t \in (0, 1/(4L))$. Then*

$$\mathbb{E}_{\boldsymbol{x}}[\Psi(\bar{\boldsymbol{\theta}}_N) - \Psi^*] \leq \frac{\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\|_2^2}{\sum_{j=0}^{N-1}\eta_j} + 2\sigma_{\Psi^*}^2\frac{\sum_{j=0}^{N-1}\eta_j^2}{\sum_{j=0}^{N-1}\eta_j} + c_Z\frac{\sum_{j=0}^{N-1}\eta_j^2\gamma_j^2}{\sum_{j=0}^{N-1}\eta_j}$$

where

$$\bar{\boldsymbol{\theta}}_N = \frac{\sum_{j=0}^{N-1}\eta_j\boldsymbol{\theta}_j}{\sum_{j=0}^{N-1}\eta_j}\ ,$$

$\sigma_{\Psi^*}^2$ is the gradient noise given in (7.17) and c_Z is given in (2.5).

Proof. The proof follows classical approach, as summarized in [23]. Let $\boldsymbol{\theta}^* \in \operatorname{argmin}(\Psi)$. Then

$$\begin{aligned}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|_2^2 &= \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^* - \eta_t\left(\nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t) + \gamma_t\boldsymbol{Z}_t\right)\|_2^2 \\ &= \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2^2 - 2\eta_t\langle\boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t) + \gamma_t\boldsymbol{Z}_t\rangle + \eta_t^2\|_2\nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t) + \gamma_t\boldsymbol{Z}_t\|_2^2\ .\end{aligned}$$

We take $\mathbb{E}_{\boldsymbol{x},t}$ (the conditional expectation with respect to data $\boldsymbol{x}$ and the current value of $\boldsymbol{\theta}_t$; see Section 7.9) and evaluate each term separately.

- Inner product term: We use (7.15) and (7.12) to get

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x},t}\left[\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t) + \gamma_t \boldsymbol{Z}_t\rangle\right] \\
&= \mathbb{E}_{\boldsymbol{x},t}\left[\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t)\rangle\right] + \mathbb{E}_{\boldsymbol{x},t}\left[\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \gamma_t \boldsymbol{Z}_t\rangle\right] \\
&= \langle \boldsymbol{\theta}^* - \boldsymbol{\theta}_t, \nabla\Psi(\boldsymbol{\theta}_t)\rangle + \gamma_t\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \mathbb{E}_{\boldsymbol{x},t}[\boldsymbol{Z}_t]\rangle = \langle \boldsymbol{\theta}^* - \boldsymbol{\theta}_t, \nabla\Psi(\boldsymbol{\theta}_t)\rangle \ .
\end{aligned}$$

- The norm term: Using (7.12), (2.5) and the fact that $\boldsymbol{Z}_t$ is zero-mean and independent of everything else,

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x},t}\left[\|\nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t) + \gamma_t \boldsymbol{Z}_t\|_2^2\right] \\
&= \mathbb{E}_{\boldsymbol{x},t}\left[\|\nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t)\|^2\right] + 2\gamma_t\mathbb{E}_{\boldsymbol{x},t}\left[\langle\nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t), \boldsymbol{Z}_t\rangle\right] + \gamma_t^2\mathbb{E}_{\boldsymbol{x},t}\left[\|\boldsymbol{Z}_t\|_2^2\right] \\
&= \mathbb{E}_{\boldsymbol{x},t}\left[\|\nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t)\|^2\right] + \gamma_t^2 c_Z \ .
\end{aligned}$$

Next, we use the convexity (7.3):

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x},t}[\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|^2] \\
&\leq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|^2 + 2\eta_t(\Psi(\boldsymbol{\theta}^*) - \Psi(\boldsymbol{\theta}_t)) + \eta_t^2\mathbb{E}_{\boldsymbol{x},t}\left[\|\nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t)\|^2\right] + \eta_t^2\gamma_t^2 c_Z \ .
\end{aligned}$$

We bound the last term by using the variance transfer of Lemma 7.9.2 (note that since we use $\mathbb{E}_{\boldsymbol{x},t}$ here, $\boldsymbol{\theta}_t$ is fixed). Using the assumption on $\eta_t L_b$ we obtain

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x},t}[\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|^2] \\
&\leq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|^2 + 2\eta_t(2\eta_t L - 1)(\Psi(\boldsymbol{\theta}_t) - \Psi(\boldsymbol{\theta}^*)) + 2\eta_t^2\sigma_{\Psi^*}^2 + \eta_t^2\gamma_t^2 c_Z \\
&\leq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|^2 - \eta_t(\Psi(\boldsymbol{\theta}_t) - \Psi(\boldsymbol{\theta}^*)) + 2\eta_t^2\sigma_{\Psi^*}^2 + \eta_t^2\gamma_t^2 c_Z \ .
\end{aligned}$$

After re-arranging the terms, taking expectation with respect to θ_t and using the telescopic sums we obtain

$$\begin{aligned}
&\sum_{j=0}^{N-1}\eta_j\mathbb{E}_{\boldsymbol{x}}\left[\Psi(\boldsymbol{\theta}_t) - \Psi(\boldsymbol{\theta}^*)\right] \\
&\leq \|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\|^2 - \mathbb{E}_{\boldsymbol{x}}[\|\boldsymbol{\theta}_N - \boldsymbol{\theta}^*\|^2] + 2\sigma_{\Psi^*,b}^2\sum_{j=0}^{N-1}\eta_j^2 + c_Z\sum_{j=0}^{N-1}\eta_j^2\gamma_j^2 \ .
\end{aligned}$$

Set $A_N = \sum_{j=0}^{N-1}\eta_j$. Divide by A_N and use convexity of Ψ to get

$$\mathbb{E}_{\boldsymbol{x}}[\Psi(\bar{\boldsymbol{\theta}}_N) - \Psi^*] \leq \sum_{j=0}^{N-1}\frac{\eta_j}{A_N}\mathbb{E}_{\boldsymbol{x}}\left[\Psi(\boldsymbol{\theta}_t) - \Psi(\boldsymbol{\theta}^*)\right] \ .$$

The result follows. □

7.8 Comments

Let us recall Lemma 6.3.1. If we have a time series with total dependence, $\boldsymbol{x}_t = \boldsymbol{x}_1$ for all t and if each query is ε-DP, at the t-th step we get εt-DP. The privacy deteriorates at each step, due to the total dependence. The SGD has a similar structure, thanks to the temporal dependence between $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{t+1}$. In other words, if we add the same noise at each iteration, we expect privacy to decrease.

Example 7.8.1. Assume that the learning rate is constant, $\eta_t = \eta$. Assume that $\beta_t = 1$, that is, we want to preserve the same ε-DP at each iteration. Then

$$\gamma_t = \frac{1}{\beta_t}\left\{\frac{\sum_{j=0}^{t-2}\eta_j}{\eta_{t-1}} + 1\right\}\Delta(\nabla\psi)/\varepsilon = t\Delta(\nabla\psi)/\varepsilon\ .$$

Thus, at each iteration t we increase the noise by a factor t.

How does this affect convergence? We analyse each term in Theorem 7.7.1 (removing all the constants).

$$\begin{aligned}
A_1 &:= \frac{1}{\sum_{j=0}^{N-1}\eta_j} = N^{-1}\ ,\\
A_2 &:= \frac{\sum_{j=0}^{N-1}\eta_j^2}{\sum_{j=0}^{N-1}\eta_j} = \eta^2/\eta\ ,\\
A_3 &:= \frac{\sum_{j=0}^{N-1}\eta_j^2\gamma_j^2}{\sum_{j=0}^{N-1}\eta_j} \approx \frac{\sum_{j=0}^{N-1} j^2}{N} \approx N^2\ ,
\end{aligned}$$

where $\approx$ means that we removed all the constants that do not depend on N. It is readily seen that $A_3 \to \infty$ as $N \to \infty$. Thus, the DP-SGD with the constant learning rate and the constant level of privacy diverges.

Example 7.8.2. In this example, we consider a decreasing learning rate, $\eta_t = (1+t)^{-\rho}$, $\rho \in (0,1)$ (a typical choice in the SGD literature). Assume again that $\beta_t = 1$, that is, we want to preserve the same ε-DP at each iteration. Then

$$\gamma_t = \frac{1}{\beta_t}\left\{\frac{\sum_{j=0}^{t-2}\eta_j}{\eta_{t-1}} + 1\right\}\Delta(\nabla\psi)/\varepsilon \approx \left\{\frac{(t-1)^{-\rho+1}}{t^{-\rho}} + 1\right\}\Delta(\nabla\psi)/\varepsilon \approx t\Delta(\nabla\psi)/\varepsilon\ .$$

Thus, again, at each iteration t we increase the noise by a factor t.

Next,

$$A_1 = \frac{1}{\sum_{j=0}^{N-1} \eta_j} = \frac{1}{\sum_{j=0}^{N-1} (j+1)^{-\rho}} \approx N^{\rho-1} \to 0 \ \text{ as } N \to \infty \,,$$
$$A_2 = \frac{\sum_{j=0}^{N-1} \eta_j^2}{\sum_{j=0}^{N-1} \eta_j} = \frac{\sum_{j=0}^{N-1} (j+1)^{-2\rho}}{\sum_{j=0}^{N-1} (j+1)^{-\rho}} \approx N^{-\rho} \to 0 \ \text{ as } N \to \infty \,,$$
$$A_3 = \frac{\sum_{j=0}^{N-1} j^{-2\rho} j^2}{\sum_{j=0}^{N-1} j^{-\rho}} \approx \frac{N^{3-2\rho}}{N^{1-\rho}} = N^{2-\rho} \to \infty \ \text{ as } N \to \infty \,.$$

Again, the DP-SGD with the decreasing learning rate and the constant level of privacy diverges.

Example 7.8.3. In this example, we consider again the decreasing learning rate, $\eta_t = (1+t)^{-\rho}$, $\rho \in (0,1)$. Now, we assume that $\beta_t = t^\kappa$, $\kappa > 0$, that is, the privacy decreases at each iteration. We want to choose κ as small as possible. Then

$$\gamma_t = \frac{1}{\beta_t} \left\{ \frac{\sum_{j=0}^{t-2} \eta_j}{\eta_{t-1}} + 1 \right\} \Delta(\nabla\psi)/\varepsilon \approx t^{-\kappa} \left\{ \frac{(t-1)^{-\rho+1}}{t^{-\rho}} + 1 \right\} \Delta(\nabla\psi)/\varepsilon \approx t^{1-\kappa} \Delta(\nabla\psi)/\varepsilon \,.$$

Next, A_1 and A_2 are the same as in the preceding example. On the other hand,

$$A_3 := \frac{\sum_{j=0}^{N-1} j^{-2\rho} j^{2-2\kappa}}{\sum_{j=0}^{N-1} j^{-\rho}} \approx \frac{N^{3-2\rho-2\kappa}}{N^{1-\rho}} = N^{2-\rho-2\kappa} \,.$$

The last term vanishes whenever $2\kappa + \rho > 2$.

In application, one often chooses $\rho = 1/2$. Then the optimal choice $\kappa > 3/4$.

If the learning rate is constant, the term A_3 vanishes whenever $\kappa > 3/2$.

7.9 Appendix: Computations for conditional expectations

We will consider different expectations $\mathbb{E}$.

- We can consider $\mathbb{E}_{\boldsymbol{x}}[\cdot] := \mathbb{E}[\cdot \mid \boldsymbol{x}]$ the conditional expectation given the data $\boldsymbol{x}$. Then, the randomness is through the sequence of indices Ω_t, $t \geq 0$ and the sequence $\boldsymbol{\theta}_t$.

- We can consider $\mathbb{E}_{\boldsymbol{x},t}[\cdot] := \mathbb{E}[\cdot \mid \boldsymbol{x}, \boldsymbol{\theta}_t]$ the conditional expectation given the data $\boldsymbol{x}$ and the current approximation $\boldsymbol{\theta}_t$ to $\boldsymbol{\theta}^*$. Then, the randomness is through the sequence of indices I_t, $t \geq 0$.

- We can consider the unconditional expectation. Then, the randomness is through both the sequence of indices I_t, $t \geq 0$ and the data $\boldsymbol{x}$.

We recall the tower property: for any random element A,

$$\mathbb{E}_{\boldsymbol{x}}[\mathbb{E}_{\boldsymbol{x},t}[A]] = \mathbb{E}_{\boldsymbol{x}}[A] \, .$$

Thus

$$\mathbb{E}_{\boldsymbol{x},t}[Z_t] = \mathbb{E}[Z_t] = \mathbf{0} \, , \quad \mathbb{E}_{\boldsymbol{x},t}[\|Z_t\|_2^2] = \mathbb{E}[\|Z_t\|_2^2] \, . \tag{7.12}$$

Formulas for expected values. We calculate

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}}\left[\widehat{\Psi}_t(\boldsymbol{\theta})\right] = \mathbb{E}_{\boldsymbol{x}}\left[\psi_{I)t}(\boldsymbol{\theta})\right] &= \mathbb{E}_{\boldsymbol{x}}\left[\sum_{j=1}^{n} 1\{I_t = j\}\psi_{I_t}(\boldsymbol{\theta})\right] \\
&= \sum_{j=1}^{n} \psi_j(\boldsymbol{\theta})\mathbb{E}_{\boldsymbol{x}}\left[1\{I_t = j\}\right] \\
&= \sum_{j=1}^{n} \psi_j(\boldsymbol{\theta})\mathbb{P}(I_t = j) \\
&= \frac{1}{n}\sum_{j=1}^{n} \psi_j(\boldsymbol{\theta}) = \Psi(\boldsymbol{\theta}) \, .
\end{aligned}$$

We used the property that $\psi_j(\boldsymbol{\theta}) = \psi(\boldsymbol{\theta}; x_j)$ is $\mathcal{X}$-measurable. Also, the conditional expectation $\mathbb{E}_{\boldsymbol{x}}$ in the second line becomes unconditional, since the index set is sampled independently of the data. Similarly,

$$\mathbb{E}_{\boldsymbol{x}}\left[\nabla\widehat{\Psi}_t(\boldsymbol{\theta})\right] = \nabla\Psi(\boldsymbol{\theta}) \, , \quad \mathbb{E}_{\boldsymbol{x}}\left[\|\nabla\widehat{\Psi}_t(\boldsymbol{\theta})\|\right] = \|\nabla\Psi(\boldsymbol{\theta})\| \, . \tag{7.13}$$

In other words, $\nabla\widehat{\Psi}_t(\boldsymbol{\theta})$ is an unbiased estimator of the gradient $\nabla\Psi(\boldsymbol{\theta})$. Furthermore,

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}}[\langle\widehat{\Psi}_t(\boldsymbol{\theta}), \Psi_t(\boldsymbol{\theta})\rangle] = \langle\Psi_t(\boldsymbol{\theta}), \Psi_t(\boldsymbol{\theta})\rangle = \|\Psi_t(\boldsymbol{\theta})\|^2 \, , \\
&\mathbb{E}_{\boldsymbol{x}}[\langle\nabla\widehat{\Psi}_t(\boldsymbol{\theta}), \nabla\Psi_t(\boldsymbol{\theta})\rangle] = \langle\nabla\Psi_t(\boldsymbol{\theta}), \nabla\Psi_t(\boldsymbol{\theta})\rangle = \|\nabla\Psi_t(\boldsymbol{\theta})\|^2 \, .
\end{aligned} \tag{7.14}$$

For the unconditional expectations we have (cf. (7.6))

$$\begin{aligned}
\mathbb{E}\left[\widehat{\Psi}_t(\boldsymbol{\theta})\right] &= \mathbb{E}\left[\mathbb{E}_{\boldsymbol{x}}\left[\widehat{\Psi}_t(\boldsymbol{\theta})\right]\right] \\
&= \frac{1}{n}\sum_{j=1}^{n} \mathbb{E}\left[\psi(\boldsymbol{\theta}; Z_j)\right] = \mathbb{E}[\Psi(\boldsymbol{\theta}; \boldsymbol{X})] = \phi(\boldsymbol{\theta})
\end{aligned}$$

The above calculations do not apply when one wants to calculate $\mathbb{E}_{\boldsymbol{x}}[\widehat{\Psi}_t(\boldsymbol{\theta}_t)]$ or $\mathbb{E}[\widehat{\Psi}_t(\boldsymbol{\theta}_t)]$. Indeed, we recall that $\boldsymbol{\theta}_t$, $t \geq 1$, is a random sequence that depends on the selected indices I_t. We can mimic computation for $\mathbb{E}_{\boldsymbol{x}}\left[\widehat{\Psi}_t(\boldsymbol{\theta})\right]$ by conditioning additionally on $\boldsymbol{\theta}_t$:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x},t}[\widehat{\Psi}_t(\boldsymbol{\theta}_t)] = \mathbb{E}[\widehat{\Psi}_t(\boldsymbol{\theta}_t) \mid \boldsymbol{x}; \boldsymbol{\theta}_t] &= \mathbb{E}\left[\sum_{j=1}^n 1\{I_t = j\}\psi_{I_t}(\boldsymbol{\theta}_t) \mid \boldsymbol{x}; \boldsymbol{\theta}_t\right] \\
&= \sum_{j=1}^n \psi_j(\boldsymbol{\theta}_t)\mathbb{E}\left[1\{I_t = j\} \mid \boldsymbol{x}; \boldsymbol{\theta}_t\right] \\
&= \sum_{j=1}^n \psi_j(\boldsymbol{\theta}_t)\mathbb{P}(I_t = j) \\
&= \frac{1}{n}\sum_{j=1}^n \psi_j(\boldsymbol{\theta}_t) = \Psi(\boldsymbol{\theta}_t)
\end{aligned}$$

and

$$\mathbb{E}_{\boldsymbol{x},t}[\nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t)] = \mathbb{E}[\nabla\widehat{\Psi}_t(\boldsymbol{\theta}_t) \mid \boldsymbol{x}; \boldsymbol{\theta}_t] = \nabla\Psi(\boldsymbol{\theta}_t)\ . \tag{7.15}$$

Formulas for variances. We have

$$\mathrm{Var}_{\boldsymbol{x}}(\nabla\widehat{\Psi}_t(\boldsymbol{\theta})) = \mathbb{E}_{\boldsymbol{x}}\left[\|\nabla\widehat{\Psi}_t(\boldsymbol{\theta}) - \nabla\Psi(\boldsymbol{\theta})\|^2\right] = \frac{1}{n}\sum_{j=1}^n \|\nabla\psi_j(\boldsymbol{\theta}) - \nabla\Psi(\boldsymbol{\theta})\|^2\ .$$

Denote

$$\sigma^2_{\Psi} := \frac{1}{n}\sum_{j=1}^n \|\nabla\psi_j(\boldsymbol{\theta}) - \nabla\Psi(\boldsymbol{\theta})\|^2\ .$$

Definition 7.9.1. *We define the* ***gradient noise*** *as*

$$\sigma^2_{\Psi^*} := \inf_{\boldsymbol{\theta}^* \in \mathrm{argmin}\Psi} \mathrm{Var}_{\boldsymbol{x}}(\nabla\widehat{\Psi}_t(\boldsymbol{\theta}))\ . \tag{7.16}$$

Note that the definition does not depend on t. In ψ_i are convex,

$$\sigma^2_{\Psi^*} = \mathrm{Var}_{\boldsymbol{x}}(\nabla\widehat{\Psi}_t(\boldsymbol{\theta}^*)) = \mathbb{E}_{\boldsymbol{x}}[\|\nabla\widehat{\Psi}_t(\boldsymbol{\theta}^*)\|^2] \tag{7.17}$$

since

$$\mathbb{E}_{\boldsymbol{x}}\left[\nabla\widehat{\Psi}_t(\boldsymbol{\theta}^*)\right] = \nabla\Psi(\boldsymbol{\theta}^*) = 0\ .$$

The next lemmas, called **variance transfer**, control the variance of the stochastic gradient at any point $\boldsymbol{\theta}$ by the corresponding variance at $\boldsymbol{\theta}^*$

Lemma 7.9.2. *Assume that ψ_i are L-smooth. Then for any $\boldsymbol{\theta} \in \mathbb{R}^p$ and any $t \in \mathbb{N}$,*

$$\mathbb{E}_{\boldsymbol{x}}\left[\|\nabla \Psi_t(\boldsymbol{\theta})\|^2\right] \leq 4L(\Psi(\boldsymbol{\theta}) - \Psi^*) + 2\sigma_{\Psi^*}^2 \,.$$

Proof. We have

$$\|\nabla \widehat{\Psi}_t(\boldsymbol{\theta})\|^2 \leq 2\|\nabla \widehat{\Psi}_t(\boldsymbol{\theta}) - \nabla \widehat{\Psi}_t(\boldsymbol{\theta}^*)\|^2 + 2\|\nabla \widehat{\Psi}_t(\boldsymbol{\theta}^*)\|^2 \,.$$

Apply $\mathbb{E}_{\boldsymbol{x}}$ to get

$$\mathbb{E}_{\boldsymbol{x}}\left[\|\nabla \widehat{\Psi}_t(\boldsymbol{\theta})\|^2\right] \leq 2\frac{1}{n}\sum_{j=1}^{n} \|\nabla \psi_j(\boldsymbol{\theta}) - \nabla \psi_j(\boldsymbol{\theta}^*)\|^2 + 2\mathbb{E}\left[\|\nabla \widehat{\Psi}_t(\boldsymbol{\theta}^*)\|^2\right] \,.$$

Apply Lemma 7.2.6 to each ψ_j to get

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}}\left[\|\nabla \widehat{\Psi}_t(\boldsymbol{\theta})\|^2\right] &\leq 2\frac{1}{n}\sum_{j=1}^{n} \|\nabla \psi_j(\boldsymbol{\theta}) - \nabla \psi_j(\boldsymbol{\theta}^*)\|^2 + 2\mathbb{E}\left[\|\nabla \widehat{\Psi}_t(\boldsymbol{\theta}^*)\|^2\right] \\
&\leq 4L_{\max}\frac{1}{n}\sum_{j=1}^{n} (\psi_j(\boldsymbol{\theta}) - \psi_j(\boldsymbol{\theta}^*) - \langle \nabla \psi_j(\boldsymbol{\theta}^*), \boldsymbol{\theta} - \boldsymbol{\theta}^* \rangle) + 2\mathbb{E}\left[\|\nabla \widehat{\Psi}_t(\boldsymbol{\theta}^*)\|^2\right] \\
&= 4L_{\max}(\Psi(\boldsymbol{\theta}) - \Psi(\boldsymbol{\theta}^*)) - \langle \nabla \Psi(\boldsymbol{\theta}^*), \boldsymbol{\theta} - \boldsymbol{\theta}^* \rangle + 2\mathbb{E}\left[\|\nabla \widehat{\Psi}_t(\boldsymbol{\theta}^*)\|^2\right] \,.
\end{aligned}$$

We conclude by $\nabla \Psi(\boldsymbol{\theta}^*) = 0$ and (7.17). □

Chapter 8

Conclusion and future direction of research

8.1 Conclusion

The goal of this thesis was to address privacy and data utility issues in the rapidly evolving field of differential privacy. By exploring differential privacy in a mathematical and statistical context, this research has made several significant contributions.

The primary outcome of this work is a unified mathematical framework for differential privacy, merging the language and vocabulary of computer science with probability and statistics. Next, novel mechanisms were proposed in order to integrate concepts from different fields (probability, statistics, data science, time series, machine learning) to provide differential privacy guarantees. We proposed a new class of sensitivity functions to address significant practical data utility challenges. We studied differential privacy in temporal dependency structures to understand the impact of privacy and utility over time, broadening the applicability of differential privacy in real-world settings.

We believe that the impact of these findings and contributions can be significant. They contribute to the existing body of knowledge by providing mechanisms to improve the utility of data. These contributions are essential for the secure and responsible use (and reuse) of data, especially in areas where sensitive data exists, such as healthcare. Moreover, the framework developed in this thesis enables researchers to study other topics in statistics with differential privacy.

Despite these contributions, this research is not without its limitations. Often, the framing of the problem has been as difficult as the proof of differential privacy guarantees. This was particularly noticeable in the time series and machine learning chapters, where complex modelling problems and temporal dependencies make the use of differential privacy more challenging. There are many avenues that future work can take to explore

the theoretical properties in complex data structures or the more practical real-world applications. These are outlined in Section 8.2.

In conclusion, this thesis helps to emphasize the importance of developing robust privacy-preserving techniques in the era of big data. The contributions made here provide a foundation for future research and practical applications, for the intersection of mathematics and statistics with privacy. And, most importantly, they contribute to further innovation in the safe and responsible use of data. In particular, some of the research in this thesis led to a US patent and to practical studies for the Office of the Privacy Commissioner of Canada.

8.2 Future work

Future work in this area can focus on several promising directions to further enhance both data utility and data privacy guarantees, in practical applications.

Combining techniques

One direction is to combine several separate techniques proposed in this thesis to improve privacy and utility guarantees. For example,

- General sensitivity methodology can be applied to study differential privacy in both time series and Stochastic Gradient Descent.
- Likewise, the Mixed Noise mechanism can be applied to time series and Stochastic Gradient Descent to minimize the amount of noise needed at each time/iteration.
- k-noise can be made differentially private by applying novel mechanisms to the group sizes, thus extending its application to benchmark against existing statistical disclosure methods while improving data utility.

Differentially private machine learning

There is a significant need to study various machine learning algorithms under differentially private constraints ([40]). Differentially private mechanisms are applied to such problems as Stochastic Gradient Descent algorithm (possibly with mini-batching), coordinate descent algorithm, expectation-maximization (EM) algorithms. The current focus is on privacy, with little emphasis on data utility (such as convergence of differentially private algorithm).

Differentially private principal component analysis

Investigating differentially private component analysis (PCA) by utilizing novel mechanisms and sensitivity functions could provide insights into the effects of adding noise to

the data versus the principal components. This approach aims to maximize data utility while maintaining privacy in these more complex scenarios.

Differentially private guarantees in Large Language Models (LLMs)

Novel mechanisms and functions can be applied in the context of LLMs to provide privacy guarantees within an LLM and ensuring outputs do not leak sensitive information.

Chapter 9

Bibliography

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, October 2016.

[2] Luk Arbuckle and Khaled El Emam. *Building an Anonymization Pipeline: Creating Safe Data*. O'Reilly Media, 2020.

[3] Marco Avella-Medina. Privacy-preserving parametric inference: A case for robust statistics. *Journal of the American Statistical Association*, 116(534):969–983, 2021.

[4] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences, 2018.

[5] Aurélien Bellet. Privacy preserving machine learning - lecture 5: Differentially private stochastic gradient descent.

[6] Devyani Biswal, Luk Arbuckle, and Rafal Kulik. The Exploration of Identifiability Through the Definition and Examination of Privacy Models. In *Privacy in Statistical Databases*, 2020.

[7] Devyani Biswal, Luk Arbuckle, and Rafał Kulik. Disclosure metrics born from statistical evaluations of data utility, 2021. Expert Meeting on Statistical Data Confidentiality (UNECE), Poznan (Poland).

[8] Devyani Biswal, Rafał Kulik, and Luk Arbuckle. Mixed noise mechanism (mnm): an approximate differentially private gaussian mechanism for low sensitivity queries. In preparation, 2023.

[9] Alberto Blanco-Justicia, David Sanchez, Josep Domingo-Ferrer, and Krishnamurty Muralidhar. A Critical Review on the Use (and Misuse) of Differential Privacy in Machine Learning. *ACM Computing Surveys*, 55(8):1–16, August 2023. arXiv:2206.04621 [cs].

[10] Peter Brockwell and Richard Davis. *Time Series: Theory and Methods*. Springer, 1991.

[11] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. arXiv, 2016.

[12] T. Tony Cai, Yichen Wang, and Linjun Zhang. The Cost of Privacy: Optimal Rates of Convergence for Parameter Estimation with Differential Privacy. *arXiv:1902.04495 [cs, stat]*, 2020.

[13] Yang Cao, Masatoshi Yoshikawa, Yonghui Xiao, and Li Xiong. Quantifying differential privacy under temporal correlations. In *Proc. Int. Conf. Data Eng.*, pages 821–832, 2017.

[14] George Casella and Roger Berger. *Statistical Inference*. Duxbury advanced series in statistics and decision sciences. Thomson Learning, 2002.

[15] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization, 2011.

[16] Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pages 1–12. Springer, 2006.

[17] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[18] Cynthia Dwork and Guy Rothblum. Concentrated differential privacy. CoRR,abs/1603.01887, 2016.

[19] Khaled El Emam and Luk Arbuckle. *Anonymizing Health Data*. O'Reilly Media, October 2013.

[20] Mark Elliot, Elaine Mackey, and Kieron O'Hara. *The anonymisation decision-making framework 2nd Edition: European practitioners' guide*. UKAN, November 2020.

[21] European Union. General data protection regulation (gdpr). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679, 2016.

[22] Liyue Fan, Li Xiong, and Vaidy Sunderam. Differentially private multi-dimensional time series release for traffic monitoring. In *27th Data and Applications Security and Privacy (DBSec), Jul 2013, Newark, NJ, United States*, pages 33–48, 2013.

[23] Guillaume Garrigos and Robert M. Gower. Handbook of convergence theorems for (stochastic) gradient methods, 2024.

[24] J.M. Gouweleeuw, P. Kooiman, L.C.R.J. Willenborg, and P.-P. de Wolf. Post randomisation for statistical disclosure control: Theory and implementation. *Journal of official Statistics*, 14(4):463, 1998.

[25] Health Information Trust Alliance. Hitrust de-identification framework, 2015.

[26] Naoise Holohan. *Mathematical Foundations of Differential Privacy*. PhD thesis, Trinity College Dublin, 2016.

[27] Naoise Holohan, Spiros Antonatos, Stefano Braghin, and Pól Mac Aonghusa. The bounded laplace mechanism in differential privacy. arXiv, 2018.

[28] Anco Hundepool, Josep Domingo-Ferrer, Laura Franconi, Sarah Giessing, Eric Schreuder Nordholt, Kevin Spicer, and Peter-Paul De Wolf. *Handbook on Statistical Disclosure Control*, 2010.

[29] Anco Hundepool, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Eric Schulte Nordholt, Keith Spicer, and Peter-Paul de Wolf. *Statistical Disclosure Control*. Wiley, 2012.

[30] International Electrotechnical Commission International Organization for Standardization. Privacy enhancing data de-identification terminology and classification of techniques. Technical report, 2018.

[31] International Electrotechnical Commission International Organization for Standardization. Iso/iec 27559:2022 information security, cybersecurity and privacy protection – privacy enhancing data de-identification framework. Technical report, 2020.

[32] Noah Johnson, Joseph P. Near, and Dawn Song. Towards practical differential privacy for sql queries. *Proc. VLDB Endow.*, 11(5):526–539, 2018.

[33] Gautam Kamath, Vikrant Singhal, and Jonathan Ullman. Private mean estimation of heavy-tailed distributions. In *Proceedings of Machine Learning Research*, volume 125, pages 1–32, 2020.

[34] Sonali Kochhar, Bartha Knoppers, Carrol Gamble, Alan Chant, Jeffrey Koplan, and Georgina S Humphreys. Clinical trial data sharing: here's the challenge. *BMJ Open*, 9(8), 2019.

[35] Ashwin Machanavajjhala, Xi He, and Michael Hay. Differential privacy in the wild: A tutorial on current practices & open challenges. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1727–1730, 2017.

[36] Paul Mangold, Aurélien Bellet, Joseph Salmon, and Marc Tommasi. Differentially private coordinate descent for composite empirical risk minimization, 2022.

[37] Michael J. Mauboussin. If you say something is likely, how likely do people think it is? *Harvard Business Review*, July 2018.

[38] Arvind Narayanan, Joanna Huey, and Edward W. Felten. *A Precautionary Approach to Big Data Privacy*, pages 357–385. Springer Netherlands, 2016.

[39] Kobbi Nissim, Sonya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. STOC 2007, 2007.

[40] National Institute of Standards and Technology. Adversarial machine learning: A taxonomy and terminology of attacks and mitigations. Technical report, 2024.

[41] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. Microsoft Tehcnical Reports, 2009.

[42] Adam Smith. Efficient, differentially private point estimators, 2008. arXiv:0809.4794v1 [cs.CR].

[43] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248, 2013.

[44] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[45] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

[46] Leon Willenborg and Ton de Waal. *Elements of statistical disclosure control*. Lecture notes in statistics. Springer, October 2000.

[47] Da Yu, Gautam Kamath, Janardhan Kulkarni, Tie-Yan Liu, Jian Yin, and Huishuai Zhang. Individual privacy accounting for differentially private stochastic gradient descent, 2023.

Index

(α, β)-admissible, 57
L-Lipschitz, 144
L-Smoothness, 144
k-PRAM, 24
k-anonymity, 22
k-noise, 25

Absolute Privacy Measure, 22
Approximate Differential Privacy, 39
    Confidence interval, 109

Blocking
    Mean, 95
    Median, 95
Bounded Laplace Mechanism, 99

Confidence Interval
    (ε, δ)-differential privacy, 109
    Differential privacy, 108
    Mean, 108
    Noisy mean, 109

Differential Privacy
    ε-differential privacy, 37
    Blocking, 92, 96
    Bounded Laplace Mechanism, 99
    Composition property, 49
    Confidence intervals, 108
    Gaussian-mechanism, 39
    Group privacy, 48
    Laplace-mechanism, 38
    Mixed Noise Mechanism, 75, 77
    Post-processing, 43
    Preservation properties, 41
    Privacy budget, 37
    Privacy loss, 38
    Smooth sensitivity, 56
    Time Series
        Independence, 121
        Total dependence, 120
    Time series, 123, 128, 129, 136
    Total dependence, 120
    Total independence, 121
    Zero-concentrated, 111
Dilation Property, 57
Disclosure Methods
    k-PRAM, 24
    k-anonymity, 22
    k-noise, 25
    Noise addition, 25
    PRAM, 23
Distance Measures
    E_γ divergence, 19
    Kullback-Leibler, 18
    Max-divergence, 18
    Rényi divergence, 18
    Total variation, 18

Gaussian Distribution, 17
General Sensitivity, 65
Global Sensitivity, 12
    Sample mean, 13
    Sample median, 13
    Sample variance, 13
Gradient Descent, 145
gradient noise, 156

Laplace Distribution, 17, 18

Learning Rate, 144
Local Sensitivity, 12
    Distance k, 60
    Sample mean, 13
    Sample median, 13
    Sample variance, 13
    Smooth bound, 56
    Violation of DP, 55

Mean
    Smooth sensitivity, 61
Mean Query, 11
    Blocking, 95
    Post-processing, 101
    Pre-processing, 101
Mean Squared Error, 52
    Laplace mechanism, 52
Median
    Smooth sensitivity, 61
Median Query, 11
    Blocking, 95
    Post-processing, 106
    Pre-processing, 106
Mixed Noise Mechanism, 75
    Algorithm, 80
    Confidence interval table, 89
    Median, 89
    Sample mean, 87
    Threshold, 77

Noise addition, 25

Output Perturbation Mechanism, 14

Post-Processing, 14
    Sample mean, 101
    Sample median, 106
    Sample variance, 102
PRAM, 23
    k-PRAM, 24
Pre-Processing, 14
    Sample mean, 101
    Sample median, 106
    Sample variance, 103
Privacy
    Event level, 119
    User level, 119

Query
    Identity, 11
    Mean, 11
    Median, 11
    Variance, 13

Response mechanism, 13

Sample Mean
    Global Sensitivity, 13
    Local Sensitivity, 13
Sample Median
    Global Sensitivity, 13
    Local Sensitivity, 13
Sample Variance
    Global Sensitivity, 13
    Local Sensitivity, 13
Sanitized Response Mechanism, 14
Sensitivity
    General, 63
    Global, 12
    Local, 12
    Weighted, 66
Sliding Property, 57
Smooth Sensitivity, 56
    Mean query, 61
    Median query, 61
Smooth sensitivity
    Differential privacy property, 58
Stochastic Gradient Descent, 147

Variance Query, 13
    Post-processing, 102
    Pre-processing, 103
Vector AutoRegressive Model, 119

Weighted sensitivity, 66

Zero-Concentrated Differential Privacy, 111
    Approximate, 112

Appendices

Appendix A

R codes

In this chapter we include the codes that are used for different numerical experiments presented in the thesis.

R Code A.0.1. This example presents R code for local sensitivity for the mean, median, and variance. This code is used in Section 5.3 and Section 5.6.

Local sensitivity for the mean:

```
Sensitivity <- function(X) { #Sensitivity function for the mean
  n=length(X)
  f = rep(0, n)
  for(i in 2:n-1){
    a = 1:(i-1)
    b = (i+1):n
    X_2 = c(X[a], X[b])
    f[i] = abs(mean(X) - mean(X_2))  }
  max(f) }
```

Local sensitivity for the median:

```
Sensitivity_med <- function(X) { #Median sensitivity function
  X <- sort(X)
  n=length(X)
  f = rep(0, n)
  for(i in 2:n-1){
    a = 1:(i-1)
    b = (i+1):n
```

```
    X_2 = c(X[a], X[b])
    f[i] = abs(median(X) - median(X_2))  }
  max(f) }
```

Local sensitivity for the variance:

```
Sensitivity_var <- function(X) { #Variance sensitivity function
  n=length(X)
  f = rep(0, n)
  for(i in 2:n-1){
    a = 1:(i-1)
    b = (i+1):n
    X_2 = c(X[a], X[b])
    f[i] = abs(var(X) - var(X_2))  }
  max(f) }
```

R Code A.0.2. This example presents R code for produce a distribution from a Laplace-Normal random variable. It also produces a plot that compares a Laplace-Normal, Laplace, and Normal densities with equal variance. This code is used in 5.4.1.

Algorithm Block DP-I:

```
library(ggplot2)
library(stats)
library(rmutil)

#write function to split blocks and add mean of laplace noise to mean
of median of blocks
# for Algorithm Block-DP I

split_noisy_median <- function(Lambda, m, n) {
  # Generate the dataset X with n random numbers uniformly distributed
   between 0 and Lambda
  # m is the number of blocks you want
  X <- runif(n, 0, Lambda)

  # Calculate the size of each block
  block_size <- ceiling(n / m)
```

```
  # Initialize a vector to store medians of each block
  medians <- numeric(m)
  noise <- numeric(m)

  # Split X into m blocks and compute median for each block
  for (i in 1:m) {
    # Calculate start and end indices of each block
    start_thr = (i-1)*Lambda/m
    end_thr <- i*Lambda/m

    # Extract the block based on threshold
    block <- X[X>= start_thr]
    block <- block[block<end_thr]
    # Calculate the median of the block and store it
    medians[i] <- median(block)
    noise[i] <- rlaplace(1,0,2*Lambda/m)
  }

  # Compute and return the mean of the medians
  mean_medians <- mean(medians)
  mean_noise <- mean(noise)

  noisy_median_DP1 = mean_medians + mean_noise
  return(noisy_median_DP1)
}

#test function for m=5,10,20
split_noisy_median(100,5,1000)
split_noisy_median(100,10,1000)
split_noisy_median(100,20,1000)

# Run the experiment
set.seed(123)  # Seed for reproducibility
Lambda <- 100
n <- 1000
m_values <- c(5,7,10,15,20)
trials <- 1000
results <- data.frame(m = integer(), MSE = numeric())
# Empty data frame for results
true_median <- Lambda / 2  # True median for U(0, Lambda)
```

```
# Perform trials and calculate MSE for each m
for (m in m_values) {
  trials_data <- replicate(trials, split_noisy_median(Lambda, m, n))
  mse <- mean((trials_data - true_median)^2)
  results <- rbind(results, data.frame(m = m, MSE = mse))
}

# Print the results table
print(results)

# Plot the results
ggplot(results, aes(x = m, y = MSE)) +
  geom_point(color = "black") +
  geom_line(color = "black", size=0.5) +# Scatter plot of MSE values
  #geom_smooth(method = "loess", se = FALSE, color = "black",
  size = 0.5) +  # Fitted curve
  #geom_hline(yintercept = true_median, linetype = "dashed",
   color = "red") +
  labs(title = "MSE for Blocking DP-I",
       x = "Number of Blocks (m)",
       y = "MSE",
       caption = "MSE for Blocking - DP-I") +
  theme_minimal()
```

R Code A.0.3. This example presents R code for produce a distribution from a Laplace-Normal random variable. It also produces a plot that compares a Laplace-Normal, Laplace, and Normal densities with equal variance. This code is used in 5.4.2.

Algorithm Block DP-II:

```
library(ggplot2)
library(stats)
library(rmutil)

#write function to split blocks for Algorithm Block-DP II
split_assign_blocks <- function(Lambda, n, m) {
  X <- runif(n, 0, Lambda)
  block_size <- floor(n / m)  # Use floor to ensure integer division
```

```
  blocks <- list()  # Initialize an empty list to store blocks

  for(i in 1:m) {
    start_index <- (i - 1) * block_size + 1
    end_index <- min(i * block_size, n)  # Ensure not to go out of
    bounds
    block_name <- paste("block", i, sep = "")
    blocks[[block_name]] <- X[start_index:end_index]
  }

  return(blocks)  # Return the list of blocks
}

compute_noisy_median <- function(Lambda, n, m) {
  # Split X into blocks
  blocks <- split_assign_blocks(Lambda, n, m)

  # Compute the median of each block
  medians <- sapply(blocks, median)

  # Calculate the average of medians
  average_median <- mean(medians)

  # Add Laplace noise to the average with parameter Lambda/m
  noisy_median <- average_median + rlaplace(1, 0, Lambda / m)

  return(noisy_median)
}

set.seed(123)  # For reproducibility
Lambda <- 100
n <- 1000
m_values <- c(5,7,10,15,20)
trials <- 1000
results2 <- data.frame(m = integer(), MSE = numeric())
true_median <- Lambda / 2

# Perform trials and calculate MSE for each m
for (m in m_values) {
  trials_data2 <- replicate(trials, compute_noisy_median(Lambda, n, m))
```

```
  mse <- mean((trials_data2 - true_median)^2)
  results2 <- rbind(results2, data.frame(m = m, MSE = mse))
}

# Plotting the MSE results
ggplot(results2, aes(x = m, y = MSE)) +
  geom_point(color = "black") +
  geom_line(color = "black", size=0.5) +
  #geom_smooth(method = "loess", se = FALSE, color = "black",
  size=0.5) +
  #geom_hline(yintercept = true_median, linetype = "dashed",
  color = "red") +
  labs(title = "MSE for Blocking DP-II",
       x = "Number of Blocks (m)",
       y = "MSE",
       caption = "MSE for Blocking - DP-II.") +
  theme_minimal()
```

R Code A.0.4. This example presents R code for produce a distribution from a Laplace-Normal random variable. It also produces a plot that compares a Laplace-Normal, Laplace, and Normal densities with equal variance. This code is used in 4.5.9.

Density of Laplace-Normal convolution:

```
#### Code for figure in 4.5.9
library(ggplot2)

n <- 1e6      # Size of simulation
mu <- 0
sigma <- 1
alpha <- 2
lambda <- 0
beta <- 2

# Generate data
# set.seed(123)
X <- rnorm(n, mu, sigma)
Y <- ifelse(runif(n, 0, alpha + beta) < alpha, alpha, -beta)
 * rexp(n) + lambda
W <- X + Y
```

```
# Plot the histogram of the Laplace-normal distribution
hist(W, freq=FALSE, breaks=200, cex.main=1, main="Laplace-Normal,
 Normal, and Laplace Distributions", xlab="Value",
 ylab="Density",ylim=c(0,0.25))

# Overlay Normal distribution with variance 9
curve(dnorm(x, mean=mu, sd=3), add=TRUE, col="blue", lwd=2)

# Overlay Laplace distribution with variance 9
dlaplace <- function(x, mu, b) {
  return (1/(2*b) * exp(-abs(x-mu)/b))
}
b <- sqrt(9/2)  # Standard deviation for Laplace is sqrt(2)*b
curve(dlaplace(x, mu, b), add=TRUE, col="red", lwd=2)

legend("topright", legend=c("Laplace-Normal",
"Normal (Variance 9)", "Laplace (Variance 9)"),
       col=c("black", "blue", "red"), lwd=2)
```

R Code A.0.5. This example presents R code for produce to the experimental analysis performed in Techniques for disclosure control. It includes code to perform k-noise and k-PRAM and includes additional codes to compare the two using kernel-density estimation. This code specifically was used for experiments presented at the 2021 Proceedings of the Joint UNECE/Eurostat Expert Meeting on Statistical Data Confidentiality. This code is used in Section 3.6, to produce figures in Section 3.8.

k-PRAM and k-Noise:

```
library(Metrics)
library(KSgeneral)
library(tidyverse)
library(dplyr)
library(ggplot2)
library(gt)
library(glue)
library(transport)
library(philentropy)
library(gridExtra)
library(ggthemes)
```

```
dm_r = read.csv("~/Downloads/dm_r.csv")
Smoking = subset(dm_r, select=c(AGE))
#Create column to index individuals
ID = c(1:659)
Smoking = cbind(ID,Smoking)

# First we need to turn age into non integer number. To each value
we will add
# a random number from (1:365)/365 and add it to the age indicated.

Smoking$DOB = rep(0,659)
for(i in 1:length(Smoking$AGE)){
  Smoking$DOB[i]=Smoking$AGE[i]+(sample(c(1:365),1)/365)
}

# We will do two approaches: one will be to set fixed buckets and
#randomize within the bucket. The second will be to add a uniform
#jitter to each age with no buckets.

# Buckets are global and fixed: [0,4],[5,9],... etc
# Since our min and max value is 22 and 77 we will only code those
buckets. Bin that defines the intervals of buckets
bin = seq(19,79, by =5)
# One way to do this is, randomly sample length(Bucket) from the bucket
#range. This will ensure that no number can jump out of the bucket
#it's in.
for(i in 1:nrow(Bucket1)){
  Bucket1$DOB[i] = runif(1,20,24)
}
for(i in 1:nrow(Bucket2)){
  Bucket2$DOB[i] = runif(1,24,29)
}
for(i in 1:nrow(Bucket3)){
  Bucket3$DOB[i] = runif(1,29,34)
}
for(i in 1:nrow(Bucket4)){
  Bucket4$DOB[i] = runif(1,34,39)
}
for(i in 1:nrow(Bucket5)){
  Bucket5$DOB[i] = runif(1,39,44)
```

```
}
for(i in 1:nrow(Bucket6)){
  Bucket6$DOB[i] = runif(1,44,49)
}
for(i in 1:nrow(Bucket7)){
  Bucket7$DOB[i] = runif(1,49,54)
}
for(i in 1:nrow(Bucket8)){
  Bucket8$DOB[i] = runif(1,54,59)
}
for(i in 1:nrow(Bucket9)){
  Bucket9$DOB[i] = runif(1,59,64)
}
for(i in 1:nrow(Bucket10)){
  Bucket10$DOB[i] = runif(1,64,69)
}
for(i in 1:nrow(Bucket11)){
  Bucket11$DOB[i] = runif(1,69,74)
}
for(i in 1:nrow(Bucket12)){
  Bucket12$DOB[i] = runif(1,74,79)
}

# Now that we have the anonymized/changed values we can store these
# values in a new dataframe.

SmokingBucket = rbind(Bucket1,Bucket2,Bucket3,Bucket4,Bucket5,Bucket6,
Bucket7,Bucket8,Bucket9,Bucket10,Bucket11,Bucket12)
dim(Smoking)
dim(SmokingBucket)

# same dimension as original. Reorder by ID number

SmokingBucket = SmokingBucket[order(SmokingBucket$ID),]

SmokingNoise= rep(0,659)
for(i in 1:659){
  SmokingNoise[i]=Smoking$DOB[i]+runif(1,-2.5,2.5)
}
```

```
Smoking_Gen = data.frame(Smoking$ID, Smoking$AGE, Smoking$DOB ,
 SmokingBucket$DOB, SmokingNoise)
names(Smoking_Gen) = c("ID","AGE","DOB","Bucket","Noise")

# FAssign Vector that assigns the Bucket number to Age anonymized
# by both methods
BinBucket = cut(Smoking_Gen$Bucket, breaks=bin)
BinNoise = cut(Smoking_Gen$Noise, breaks=bin)

Smoking_Gen$BinBucket = tapply(Smoking_Gen$Bucket, BinBucket)
Smoking_Gen$BinNoise = tapply(Smoking_Gen$Noise,BinNoise)
# Reorder to place Bin after Randomized Age
Smoking_Gen = Smoking_Gen[c(1,2,3,4,6,5,7)]

Smoking_Gen$BucketJump = Smoking_Gen$BinBucket - Smoking_Gen$BinNoise

# Measure the bias between Bucketed Ages and Noise Ages against DOB
# Similar = dataSimilarity(SmokingO, SmokingB,dropDiscrete = NA)
bias_bucket = bias(Smoking_Gen$DOB, Smoking_Gen$Bucket)
mse_bucket = mse(Smoking_Gen$DOB, Smoking_Gen$Bucket)
rmse_bucket = rmse(Smoking_Gen$DOB, Smoking_Gen$Bucket)

bias_noise = bias(Smoking_Gen$DOB, Smoking_Gen$Noise)
mse_noise = mse(Smoking_Gen$DOB, Smoking_Gen$Noise)
rmse_noise = rmse(Smoking_Gen$DOB, Smoking_Gen$Noise)

diff_bucket = rep(0,659)
for (i in 1:659){
  diff_bucket[i] = Smoking_Gen$Bucket[i]-Smoking_Gen$DOB[i]
}

diff_noise = rep(0,659)
for (i in 1:659){
  diff_noise[i] = Smoking_Gen$Noise[i]-Smoking_Gen$DOB[i]
}

differences = data.frame(ID,diff_bucket,diff_noise)
names(differences) = c("ID","Bucket","Noise")
plot(diff_noise)
```

```
scatterPlot <- ggplot(differences,aes(x = ID)) +
  geom_point(aes(y=diff_bucket), color="black") +
  geom_point(aes(y=diff_noise),color = "red") +
  labs(x="ID", y="Change in Age",title="Scatterplot: Change in Age")+
  theme(panel.background = element_blank())

# Amount of people in each persons bucket where they are
centered at +-2.5 around them. Theoretical group size
# ExpBucket = rep(0,659)
# for(i in 1:659){
#   m = Smoking_Gen$DOB[i]
#   ExpBucket[i] = nrow(filter(Smoking_Gen,
# Smoking_Gen$DOB >= m-2.5, Smoking_Gen$DOB <= m+2.5))
# }
# plot(ExpBucket)

################################################################
# MC of expected number of people in theoretical bucket
# How noise affects histogram of data

binsizes = c(rep(0,659)) ;
data_noise = c(rep(0,659)) ;
N=1000
for(n in 1:N)
{
  Noise = runif(659,-2.5,2.5)
  SmokingNoise= Smoking_Gen$DOB + Noise
  data_noise = data_noise + SmokingNoise

  ExpBucket = rep(0,659)
  for(i in 1:659){
    q = Smoking_Gen$DOB[i]

    ExpBucket[i] = length(which(SmokingNoise>=q-2.5 &
    SmokingNoise<= q+2.5))
  }

  currentbucketsize = ExpBucket

  binsizes = binsizes + currentbucketsize
```

```
}
# End of the MC loop

binsizes = binsizes/N
data_noise = data_noise/N

binsizes_round = round(binsizes)
par(mfrow=c(1,1))
hist(Smoking_Gen$DOB,
     breaks = bin,
     main="Histogram: Original Data",
     xlab="Age",
     border="black",
     col="white",
     xlim=c(10,80))
hist(data_noise,
     breaks = bin,
     main="Histogram: Noisey Data",
     xlab="Age",
     border="black",
     col="white",
     xlim=c(10,80))
hist(Smoking_Gen$Bucket,
     breaks = bin,
     main="Histogram: Grouped Data",
     xlab="Age",
     border="black",
     col="white",
     xlim=c(10,80))

binned = cut(Smoking_Gen$Noise, breaks=(c(19,24,29,34,39,44,49,54,59,
                          64,69,74,79)))
Bin2 = tapply(Smoking_Gen$Noise,binned)
Bin2 <- as.factor(Bin2)
table(Bin2)
# Bin2
# 1   2   3   4   5   6   7   8   9  10  11  12
# 39  96 119  72  73  77  74  64  31  11   2   1

a1 <- data.frame(Bucket=c("1","2","3","4","5","6","7","8","9","10",
```

```
"11","12"), values=c(32, 100, 121,81,62,79,84,57,31,9,2,1))
b1 <- data.frame(Bucket=c("1","2","3","4","5","6","7","8","9",
"10","11","12"), values=c(39,  96, 119, 72, 73, 77, 74, 64,31, 11,2,1))

a1$Bucket <- factor(a1$Bucket, # Change ordering manually
levels = c("1","2","3","4","5","6","7","8","9","10","11","12"))
b1$Bucket <- factor(b1$Bucket, # Change ordering manually
 levels = c("1","2","3","4","5","6","7","8","9","10","11","12"))

ggplot(a1, aes(x=Bucket, y=values, sort=FALSE))+
  geom_bar(stat = "identity") +
  labs(title = "Method: k-PRAM") +
  geom_text(aes(label = values),
            position = position_dodge(0.9), vjust = -0.5,
            check_overlap = TRUE) +
  xlab("Age") +
  ylab("Frequency") +
  theme_tufte()

ggplot(b1, aes(x=Bucket, y=values, sort=FALSE))+
  geom_bar(stat = "identity") +
  labs(title = "Method: k-noise") +
  geom_text(aes(label = values),
            position = position_dodge(0.9), vjust = -0.5,
            check_overlap = TRUE) +
  xlab("Age") +
  ylab("Frequency") +
  theme_tufte()

# Expected number of people in theoretical group per individual
# data point
GroupSize = data.frame(Smoking_Gen$DOB,binsizes_round)
names(GroupSize) = c("DOB","GroupSize")

ggplot(GroupSize,aes(x = DOB))+
  geom_histogram(color="black", fill="white",binwidth = 5 ,
   breaks=c(19,24,29,34,39,44,49,54,59,64,69,74,79) )+
  geom_point(data=GroupSize,aes(x=DOB,y=GroupSize)) +
  labs(x="Age", y="Frequency ",
  title="Expected Number of Ages in Localized Group")+
```

```
 theme_tufte()

# ggplot(Age2,aes(x=Generalized)) +
#   geom_histogram(color="black", fill="white",binwidth = 5 ,
breaks=c(19,24,29,34,39,44,49,54,59,64,69,74,79) ) +
#   geom_point(data=OrderedExpB,aes(x=Age,y=Noise)) +
#   labs(x="Age", y="Frequency ",title="Expected Number of
#   Ages in Localized Group")+
#   theme_tufte()

###########Local noise distribution ##################
# Idea is to add noise inversely proportionate to the
# group size the datapoint belongs to
# Example: Individual 9: DOB GroupSize
#  9 52.72329        80
# So the noise added to that individual will be uniform(+- constant/80)
# perhaps the constant can be the mean of the binsizes
#in this iteration: 81 mean(binsizes)
#[1] 81.40651

# Smoking_MovingNoise = rep(0,659)
# for(i in 1:659){
# t = binsizes_round[i]
# Noise = runif(1,-meanbin/t,meanbin/t)
# Smoking_MovingNoise[i] = Smoking_Gen$DOB[i]+ Noise
# }

# Problem occurring at extreme end when only 1-5 people in buckets
# adds way to much noise. Need to impose condition on the noise,
# min(meanbin/t and 5)
# Smoking_MovingNoise[337]
# [1] 144.6984
# > Smoking_Gen$DOB[337]
# [1] 76.63014

# Make table summarizing these results
summ1 <- tibble(Method = c("k-PRAM","k-Noise"),
  Bias = c(bias_bucket,bias_noise),Mse = c(mse_bucket,mse_noise),
  Rmse = c(rmse_bucket,rmse_noise))
summ1 %>%
```

```
  gt() %>%
  cols_align(
    align =  "center",
    columns = everything()
  )%>%
  tab_style(
    style = list(
      cell_text( weight = "bold")
    ),
    locations = list(
      cells_column_labels(gt::everything())
    )
  ) %>%
  tab_header(
    title = "Summary of Utility Estimators"
  )

#########Some statistical tests ########################
kern.dens.DOB = density(Smoking_Gen$DOB,kernel = c("gaussian"))
kern.dens.Bucket = density(Smoking_Gen$Bucket,kernel = c("gaussian"))
kern.dens.Noise = density(Smoking_Gen$Noise,kernel = c("gaussian"))

kern.test.DOB = kern.dens.DOB$y
kern.test.Bucket = kern.dens.Bucket$y
kern.test.Noise = kern.dens.Noise$y

par(mfrow=c(1,2))
plot(kern.test.DOB , type = "l", col = "blue")
lines(kern.test.Bucket,col="green")
plot(kern.test.DOB , type = "l", col = "blue")
lines(kern.test.Noise,col="red")

densities = data.frame(dens = c(kern.test.DOB, kern.test.Bucket,
 kern.test.Noise)
 , lines1 = rep(c("DOB", "k-PRAM","k-Noise"), each = 512))
densities1 = data.frame(dens = c(kern.test.DOB, kern.test.Bucket)
 , lines1 = rep(c("DOB", "k-PRAM"), each = 512))
densities2 = data.frame(dens = c(kern.test.DOB,kern.test.Noise)
, lines2 = rep(c("DOB","k-Noise"), each = 512))
```

```
#Plots
g1 = ggplot(densities1, aes(x = dens, fill = factor(lines1))) +
  geom_density(alpha = 0.5)+
  scale_fill_manual( values = c("red","blue"))+
  labs(x="x", y=" Density ",title="Kernel Density Estimation")+
  theme_tufte()
g2 = ggplot(densities2, aes(x = dens, fill = factor(lines2))) +
  geom_density(alpha = 0.5)+
  scale_fill_manual( values = c("blue","green")) +
  labs(x="x", y=" Density ",title="Kernel Density Estimation")+
  theme_tufte()
g = ggplot(densities, aes(x = dens, fill = factor(lines1))) +
  geom_density(alpha = 0.5)+
  scale_fill_manual( values = c("red","blue","green","orange"))

grid.arrange(g1,g2)

wasserstein1d(kern.test.DOB,kern.test.Bucket,p=2)
wasserstein1d(kern.test.DOB,kern.test.Noise,p=2)
wasserstein1d(kern.test.DOB,kern.test.NoiseMove,p=2)
wasserstein1d(kern.test.DOB,kern.test.DOB,p=1)
```

R Code A.0.6. This example presents R code for statistics generated from pre-processing and post-processing differentially private mechanisms. It also produces the plot for comparing the results for the median for DOB. This code is used to produce Figure 5.15, in Section 5.6.

Median estimator: Pre-processing vs Post-processing

```
#Importing dataset and randomizing age to DOB
dm_r <- read_csv("~/Downloads/dm_r.csv")
Smoking = subset(dm_r, select=c(AGE)) #Create column to
index individuals
ID = c(1:659)
Smoking = cbind(ID,Smoking) # Create the "original" dataset that will
 be used to compare against each # anonymized dataset.
Smoking$DOB = rep(0,659); #how to randomize age
for(i in 1:length(Smoking$AGE)){
  Smoking$DOB[i]=Smoking$AGE[i]+(sample(c(1:365),1)/365) }
```

```
#Pre-processing
library(VGAM)
range_smoke.g <- 110-0
epsilon = 1

mean_DOB_pre.g = NULL ;
median_DOB_pre.g = NULL ;
var_DOB_pre.g = NULL ;

for (i in 1:1000){
  noise_DOB.g <- rlaplace(659,0,range_smoke.g/(epsilon))
  DOB_anon_pre.g <- Smoking$DOB + noise_DOB.g

  mean_DOB_pre.g[i] <- mean(DOB_anon_pre.g)
  median_DOB_pre.g[i] <- median(DOB_anon_pre.g)
  var_DOB_pre.g[i] <- var(DOB_anon_pre.g)
}

#Post-processing
##MEAN
sen.DOB.g <- 110/659
eDP.DOB.g <- function(data,e){ #Function that adds noise
  return(mean(data)+rlaplace(1,0,sen.DOB.g/e))
}

mean_DOB_post.g = NULL;
for (i in 1:1000) {
  mean_DOB_post.g[i] <- eDP.DOB.g(Smoking$DOB,1)
}

##MEDIAN
range_med_DOB.g <- abs(median(c(0:109))-median(c(1:110)))

eDP.DOB.g_med <- function(data,e){ #Function that adds noise
  return(median(data)+rlaplace(1,0,range_med_DOB.g/e))
}

median_DOB_post.g = NULL;
for (i in 1:1000) {
  median_DOB_post.g[i] <- eDP.DOB.g_med(Smoking$DOB,1)
```

```
}

##Variance
range_var_DOB.g <- c(0:110)

var.DOB = NULL;
for (i in 1:111){
  var.DOB[i] <- abs(var(range_var_DOB.g)-var(range_var_DOB.g[-i]))
}
max(var.DOB)

sen.DOB.g_var <- max(var.DOB)/(659)
eDP.DOB.g_var <- function(data,e){ #Function that adds noise
  return(var(data)+rlaplace(1,0,sen.DOB.g_var/e))
}

var_DOB_post.g = NULL;
for (i in 1:1000) {
  var_DOB_post.g[i] <- eDP.DOB.g_var(Smoking$DOB,1)
}

#### PLOTS
library(ggplot2)
library(reshape2)
library(ggthemes)

DOB.median.g <- data.frame(median_DOB_pre.g, median_DOB_post.g)
DOB.median.g$ID <- c(1:1000)
colnames(DOB.median.g) <- c("Pre-processing", "Post-processing",
"ID")
DOB.median.m.g <- melt(DOB.median.g, id.vars = "ID",
measure.vars = c("Pre-processing", "Post-processing"))

scatterplot_DOB.median.g <- ggplot(DOB.median.m.g,
aes(ID, value, colour = variable, shape = variable)) +
  geom_point(shape = 16, alpha = 0.5) +
  # Using shape = 16 for circles and setting alpha for 50% opacity
  ggtitle("Pre vs Post-processing for the median - DOB") +
  scale_color_manual(values = c("black", "#CE8BDC")) +
  labs(colour = "Pre vs Post-processing",
```

```
  shape = "Pre vs Post-processing") +
  xlab("Number of iterations") +
  ylab("Median") +
  geom_hline(data = Smoking, aes(yintercept = median(DOB)),
  color = "red") +
  theme_tufte()

scatterplot_DOB.median.g
```

R Code A.0.7. This example presents R code for all the plots produced in Chapter 6.

Experiments for Privacy Leakage in Vector Autoregressive models:

```
# Define all variables
# Case 1: epsilon_2' is a function of rho.
library(ggplot2)
library(ggthemes)
epsilon_2 = 1 #epsilon_2 original approx dp formulation
deltaf = 0.5 # sensitivity of f - fixed from 1 to 1/2
a_11 = 1
a_12 = 1


delta = 0.05
var_1 = 0.25 # variance of B_1/2
var_2 = 0.25 # variance of B_2/2
rho = seq(-1,1,0.001)

varepsilon_2 = 1/(1+((1+rho)/4*log(1.25/0.05)))^.5

epsilon.plot = data.frame(rho,varepsilon_2)

g = ggplot(epsilon.plot,aes(x=rho,y=varepsilon_2))+
  geom_point(color="darkblue",size = 0.5)+
  labs(x = expression(rho),y=expression(epsilon[2]*"'"),)+
  theme(axis.text.x = element_text(size = 10),
        axis.text.y = element_text(size = 10),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20))+
```

```
  theme_hc()

print(g + ggtitle("A1+N3"))
############ A2+N1 Experiment v1.1 ################

A<- matrix(c(0.5,0,0.1,0.5),2,2)

det(A)
C<-solve(A)
c_11 = C[1,1]
c_12 = C[1,2]
c_21 = C[2,1]
c_22 = C[2,2]

deltg = (c_11+c_21)/2
delta = 0.05
deltaf = 0.5
epsilon=1
var_1 = 1
var_2 = 1
n=2

rho2 = seq(-1,1,by=0.01)
s1 = c_11+c_21
s2 = c_12+c_22
var_sb = s1^2+s2^2+2*s1*s2*rho2

epsilon1 = deltg/(1+(var_sb)/(2*n*log(1.25/delta)))^0.5

epsilon.plot2 = data.frame(rho2,epsilon1)
g1.1 = ggplot(epsilon.plot2,aes(x=rho2,y=epsilon1))+
  geom_point(color="darkblue",size = 0.5)+
  labs(x = expression(rho),y=expression(epsilon[1]*"'"),)+
  theme(axis.text.x = element_text(size = 10),
        axis.text.y = element_text(size = 10),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20)) +
  theme_hc()
print(g1.1 + ggtitle("A2+N1 1.1"))
```

```
######## A2+N1 Experiment v1.2 #####
A<- matrix(c(0.9,0,0.5,0.9),2,2)

det(A)
C<-solve(A)
c_11 = C[1,1]
c_12 = C[1,2]
c_21 = C[2,1]
c_22 = C[2,2]

deltg = (c_11+c_21)/2
delta = 0.05
deltaf = 0.5
epsilon=1
var_1 = 1
var_2 = 1
n=2

rho2 = seq(-1,1,by=0.01)
s1 = c_11+c_21
s2 = c_12+c_22
var_sb = s1^2+s2^2+2*s1*s2*rho2

epsilon1 = deltg/(1+(var_sb)/(2*n*log(1.25/delta)))^0.5

epsilon.plot2 = data.frame(rho2,epsilon1)
g1.2 = ggplot(epsilon.plot2,aes(x=rho2,y=epsilon1))+
  geom_point(color="darkblue",size = 0.5)+
  labs(x = expression(rho),y=expression(epsilon[1]*"'"),)+
  theme(axis.text.x = element_text(size = 10),
        axis.text.y = element_text(size = 10),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20)) +
  theme_hc()
print(g1.2 + ggtitle("A2+N1 1.2"))


######### experiment 3 for A2N1 #########
####### Fix rho = -1,0,1 #####
a_11 = seq(0.1,0.9,by=0.1)
```

```
a_22 = seq(0.9,0.1,by=-0.1)
det_A = NULL;
c_11 = NULL;
c_12 = NULL;
c_21 = NULL;
c_22 = NULL;
s1 = NULL;
s2 = NULL;
var_sb = NULL;
deltaG = NULL;
epsilon1 = NULL;
rho = 1;
epsilon = 1;
var_1 = 1;
var_2 = 1;
delta = 0.05;
deltaf = .5;  #fixed from 1 to 0.5
n=2;

for(i in 1:9){
  A = matrix(c(a_11[i],0,0.7,a_22[i]),2,2)
  C = solve(A)
  c_11[i] = C[1,1]
  c_12[i] = C[1,2]
  c_21[i] = C[2,1]
  c_22[i] = C[2,2]
  deltaG[i] = (c_11[i]+c_21[i])/2
  s1[i] = c_11[i]+c_21[i]
  s2[i] = c_12[i]+c_22[i]
  var_sb[i] = (s1[i])^2+(s2[i])^2+2*s1[i]*s2[i]*rho
  epsilon1[i] = deltaG[i]/(1+(var_sb[i])/(2*n*log(1.25/delta)))^0.5

}


epsilon.plot2 = data.frame(deltaG,epsilon1)
g2.2 = ggplot(epsilon.plot2,aes(x=deltaG,y=epsilon1))+
  geom_line(color="darkblue",size = 1)+
  labs(x = "Delta G",y=expression(epsilon[1]*"'"),)+
  theme( axis.text.x = element_text(size = 10),
```

```
         axis.text.y = element_text(size = 10),
        axis.title.x = element_text(size = 10),
        axis.title.y = element_text(size = 20))+
  theme_hc()

print(g2.2 + ggtitle("A2+N1 Part 2.2"))

###### experiment 2.1 for A2N1 ##############
######### Fix rho = -1,0,1 ########
a_11 = seq(0.1,0.9,by=0.1)
a_22 = a_11
det_A = NULL;
c_11 = NULL;
c_12 = NULL;
c_21 = NULL;
c_22 = NULL;
s1 = NULL;
s2 = NULL;
var_sb = NULL;
deltaG = NULL;
epsilon1 = NULL;
rho = 1;
epsilon = 1;
var_1 = 1;
var_2 = 1;
delta = 0.05;
deltaf = .5;  #fixed from 1 to 0.5
n=2;

for(i in 1:9){
  A = matrix(c(a_11[i],0,0.7,a_22[i]),2,2)
  C = solve(A)
  c_11[i] = C[1,1]
  c_12[i] = C[1,2]
  c_21[i] = C[2,1]
  c_22[i] = C[2,2]
  deltaG[i] = (c_11[i]+c_21[i])/2
  s1[i] = c_11[i]+c_21[i]
  s2[i] = c_12[i]+c_22[i]
  var_sb[i] = (s1[i])^2+(s2[i])^2+2*s1[i]*s2[i]*rho
```

```
  epsilon1[i] = deltaG[i]/(1+(var_sb[i])/(2*n*log(1.25/delta)))^0.5

}


epsilon.plot2 = data.frame(deltaG,epsilon1)
g2.1 = ggplot(epsilon.plot2,aes(x=deltaG,y=epsilon1))+
  geom_line(color="darkblue",size = 1)+
  labs(x = "Delta G",y=expression(epsilon[1]*"'"),)+
  theme( axis.text.x = element_text(size = 10),
         axis.text.y = element_text(size = 10),
         axis.title.x = element_text(size = 10),
         axis.title.y = element_text(size = 20))+
  theme_hc()

print(g2.1 + ggtitle("A2+N1 2.1"))
```