On the Trade-Off between Privacy and Utility in Statistical Data Analysis

Dissertation

Zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
der Ruhr-Universität Bochum
an der Fakultät für Mathematik
am Lehrstuhl Mathematik & Informatik

vorgelegt von
Francesco Aldà

unter der Betreuung von
Prof. Dr. Hans Ulrich Simon

Bochum, Oktober 2017
Francesco Aldà

*On the Trade-Off between Privacy and Utility in Statistical Data Analysis*

October 2017

First reviewer: Prof. Dr. Hans Ulrich Simon
Second reviewer: Prof. Dr. Alexander May

**Ruhr-Universität Bochum**
Fakultät für Mathematik
Lehrstuhl Mathematik & Informatik
Universitätsstraße 150
44801 Bochum
Abstract

In this thesis we investigate the trade-off that results from combining the conflicting goals of performing accurate statistical analysis on sensitive data while providing strong privacy guarantees. Specifically, we restrict our attention to the paradigm of differential privacy [30] and consider a range of challenging problems related to data analytics and machine learning. The main contributions of this work can be summarized as follows.

(i) We consider the problem of releasing integer partitions under differential privacy and provide a lower bound on the minimax risk that any differentially private mechanism must incur.

(ii) We provide general conditions under which the expected error introduced by the exponential mechanism [66], a well-established tool for providing differential privacy, is optimal.

(iii) We investigate the problem of function release under differential privacy. We propose a generic mechanism—we name the Bernstein mechanism—and establish fast convergence rates on the error it introduces. We also provide lower bounds on the error that any differentially private functional mechanism must incur.

(iv) Finally, we consider a distributed setting where no trusted entity can be relied upon and the users are responsible for protecting the privacy of their own data. We then provide some conditions under which differentially private randomized response schemes enable to infer (in probability) an approximation of the true fraction of instances in a database satisfying a given predicate. On the contrary, when these conditions do not hold, we show that the strong requirements of differential privacy rule out any meaningful result.
Acknowledgments

First and foremost, I must thank my advisor, Prof. Hans Ulrich Simon, who introduced me to the very exciting world of differential privacy, taught me how to be a good mathematician and researcher, and gave me the freedom to follow my interests and passions. I feel very fortunate for his guidance and support over the last four years.

I would like to express my gratitude to Prof. Benjamin I. P. Rubinstein, with whom I had the honour and pleasure to spend two wonderful months at the University of Melbourne. I learned a great deal and had a really amazing time. Sharing ideas and thoughts while sipping some delicious coffee in the campus is something that I will never forget!

I thank my colleagues from LMI and UbiCrypt at the Ruhr-Universität Bochum for the bliss and support I always felt with them and for the endless patience they showed when listening to my poor German. Special thanks go to my office mates, Malte and Filipp, for making our office an enjoyable place to work in, and for the infinite brainstorming sessions we had together.

I am much indebted with Davide and Luca for sharing joy and melancholy, and for making this journey in Bochum one of the most unforgettable experiences of my life.

I would like to thank my family for the firm belief in me and the steady encouragements. Even when the difficulties looked overwhelming, feeling their presence kept me going on.

Last but not least, my deepest thanks go to Irene for simply being the most extraordinary creature on Earth. She has always got my back, no matter what. I would never have achieved this goal without her. Grazie!
Contents

1 Introduction ................................. 1
  1.1 Organization of the Thesis .............. 2

2 Preliminaries .............................. 5
  2.1 Notation and Useful Results .......... 5
  2.2 Learning Theory ......................... 7
    2.2.1 Statistical Query Model .......... 8
    2.2.2 Linear Arrangements and Margin Parameters ...... 10

3 Differential Privacy ....................... 13
  3.1 Motivation and Informal Discussion ... 13
  3.2 Definition and Properties ............. 14
  3.3 Basic Tools ............................. 16
    3.3.1 Laplace Mechanism ................. 17
    3.3.2 Exponential Mechanism .......... 18

4 A Lower Bound on the Release of Integer Partitions 19
  4.1 Introduction ............................. 19
  4.2 Setting .................................. 20
  4.3 Lower Bound Based on Assouad’s Lemma ...... 22
  4.4 Related Work ............................. 25
  4.5 Conclusions and Open Problems ......... 25

5 Average-case Optimality of the Exponential Mechanism 27
  5.1 Introduction ............................. 27
  5.2 Optimal Mechanisms and Linear Programming .......... 28
  5.3 Proof of Claim 5.4 and Additional Remarks on LP** ........ 34
  5.4 A Toy Example: the Path Graph ............ 39
  5.5 Related Work ............................. 44
  5.6 Conclusions and Open Problems .......... 45
## Contents

6 Function Release and the Bernstein Mechanism 47  
6.1 Introduction ........................................... 47  
6.2 Setting ................................................. 48  
6.3 The Bernstein Mechanism ................................. 49  
6.4 Privacy and Utility Analysis ............................... 51  
  6.4.1 Proof of Theorem 6.2 for $\ell = 1$ .................. 52  
  6.4.2 Proof of Theorem 6.2 ............................... 55  
  6.4.3 Discussion ......................................... 58  
6.5 Lower Bound ............................................ 59  
6.6 Examples ............................................... 60  
6.7 Related Work .......................................... 66  
6.8 Conclusions and Additional Remarks ..................... 66  

7 Randomized Response Schemes and the SQ-model 69  
7.1 Introduction ........................................... 69  
7.2 Setting ................................................. 71  
7.3 Outline of the Main Results ............................... 72  
7.4 Proof of Theorem 7.1 for Balanced Classes ................ 74  
7.5 Proof of Theorem 7.1 .................................. 79  
7.6 Proof of Theorem 7.2 .................................. 82  
7.7 Arrangements with a Non-Negligible Minimum Margin ...... 88  
7.8 Efficiency Issues ...................................... 92  
7.9 Related Work .......................................... 94  
7.10 Conclusions .......................................... 96  

Bibliography ................................................. 97  

List of Symbols ............................................ 105  

List of Figures ............................................. 109
Introduction

Statistical data analysis, and more specifically machine learning, is one of the most flourishing areas of modern mathematics and computer science. Both terms are used to identify the field which is concerned with the study and the design of algorithms that can inspect, transform and model data in order to extract useful information which can later be used to support decision-making or predict the outcome of unknown events. Machine learning based technology has become very common in our daily life. Anti-spam software which learns to filter out undesired messages from our inbox, digital cameras which learn to detect faces and label objects each time we shoot a photograph, and recommendation systems which suggest us items we may be interested in buying based on our previous purchases are just a few examples. This area has seen an unprecedented development over the last few decades, mainly due to the impact and the potentiality of the applications which can be derived, the availability of large datasets, and the constantly increasing amount of information that can be collected and curated every day. As these capabilities permit to access information about individuals at a very fine-grained level, serious concerns about privacy violation arise, especially when sensitive data, e.g., medical records or user profiles, are involved or when untrusted parties are the recipients of such analyses. As a consequence, there is a fundamental need for providing robust, and mathematically rigorous privacy guarantees to the individuals whose data are collected and processed. Differential privacy \cite{Dwork2006} has recently emerged as the de facto framework for privacy-preserving statistical data analysis (also referred to as private data analysis). It formally ensures privacy by limiting the influence of an individual record on the released statistics or analysis. Research on differential privacy is largely devoted to the development and study of mechanisms, i.e., randomized algorithms, that satisfy this privacy definition while enabling a rich class of statistical analyses and machine learning algorithms to be performed.
The contents of this thesis represent novel results which fit within this research area. More specifically, this work investigates the trade-off that naturally arises when trying to combine the demands of differential privacy with the capability of extracting useful global information from data. This challenging problem is commonly referred to as the trade-off between privacy and utility. In other words, we aim to study the “quality” of the results of a privacy-preserving statistical analysis, if any can be achieved at all, by comparing them to those we would obtain when neglecting privacy issues. This research topic has attracted a great deal of attention in many fields, from statistics and theoretical computer science to security. In our treatment, we will be using a variety of techniques which are proper of each one of these fields.

Even though privacy can be defined in very generic terms as a statistical property that an algorithm should possess, utility does depend on the specific task or analysis we are interested in performing. Therefore, each chapter will be addressing a peculiar problem, e.g., from the very specific task of privately releasing integer partitions (Chapter 4) to the more challenging goal of answering counting queries in a distributed setting (Chapter 7). Nevertheless, the objectives of this thesis are twofold. First, we aim to provide bounds on the error that any differentially private mechanism must incur when solving a specific task, and analyze the optimality of mechanisms that are by now regarded to as state-of-the-art tools for achieving differential privacy. Answering this kind of questions is vital in order to shed some light on the hardness of providing differential privacy and assuring the best utility guarantees for the results which are delivered. Second, we wish to develop new mechanisms extending the capabilities of current differentially private mechanisms, so as to enrich the classes of statistical analyses for which high-quality results can be obtained while providing differential privacy.

1.1 Organization of the Thesis

This work can be divided into four parts.

(i) The first part provides a gentle introduction to the main topics discussed in this thesis. It consists of Chapter 2 which presents some notation and recall well-known tools and results from different areas of mathematics, and Chapter 3 which provides a more thorough introduction to differential privacy and its promise. In particular, Chapter 2 defines the notation and presents relevant results on learning theory, i.e., the study of machine learning from a theoretical and mathematically rigorous point of view. Our attention will notably be paid on the statistical query model of learning. Chapter 3 motivates the need for a robust framework for privacy-preserving statistical analysis, introduces the definition and the
main properties of differential privacy, and presents the most renowned building blocks used for providing it, namely the Laplace and the exponential mechanism.

(ii) The second part consists of Chapters 4 and 5. In Chapter 4 we consider the problem of releasing integer partitions under differential privacy and provide a lower bound on the minimax risk that any differentially private mechanism releasing an integer partition must incur. Our lower bound is optimal (up to a constant factor) for small integers. For large values, it does not match the upper bound shown by Blocki et al. [12], achieved by the exponential mechanism (and its efficient implementation). Chapter 5 tries to close this gap by tackling the problem from a different perspective. It more generally studies the optimality of the error introduced by the exponential mechanism in the average-case scenario. The results presented provide general conditions under which the exponential mechanism is optimal and shed some light on how it might be possible to show its (average-case) optimality for the specific task of releasing integer partitions.

(iii) In Chapter 6 we investigate function release under differential privacy. The goal is to provide a generic mechanism that is able to “sanitize” functions and applies under the weak assumptions of oracle access to target function evaluation and sensitivity, akin to the Laplace mechanism whose applicability is restricted to vector-valued functions only. We propose a mechanism, termed the Bernstein mechanism, and establish upper bounds on the error it incurs. We also provide lower bounds on the utility achievable for any functional mechanism that provides differential privacy. We moreover demonstrate the generality and versatility of our proposed solution by presenting and analyzing a number of examples.

(iv) The last part of this work consists of Chapter 7. In contrast to Chapters 5 and 6, we focus on a more challenging setting where each user is responsible for protecting the privacy of her own data and no trusted party or any communication with other entities is involved. The mechanisms employed in such a setting are referred to as randomized response schemes. We introduce the notions of weak and strong usefulness (capturing an average-case scenario) and investigate under which conditions a differentially private randomized response scheme is strongly useful or, in contrast, differential privacy rules out even weak usefulness. By doing so, we establish strong connections between differential privacy and the statistical query model of learning, which is a quite influential model in machine learning theory.

1This is also known as the local model of privacy.
Chapter 1. Introduction

We conclude this section highlighting that this thesis is based on the following peer-reviewed and previously published works, which contain most of the results discussed here:

- “A Lower Bound on the Release of Differentially Private Integer Partitions” [5], by the author and Hans Ulrich Simon;
- “On the Optimality of the Exponential Mechanism” [4], by the author and Hans Ulrich Simon;
- “The Bernstein Mechanism: Function Release under Differential Privacy” [2], by the author and Benjamin I. P. Rubinstein;
- “Randomized Response Schemes, Privacy and Usefulness” [3], by the author and Hans Ulrich Simon.
Chapter 2

Preliminaries

We first introduce some notation and basic facts which are used throughout this work. For the sake of completeness, we refer the reader to the List of Symbols which provides a summary of the relevant notation employed in this thesis. We then give a gentle introduction to learning theory, with a special focus on the statistical query model of learning and its connections to linear arrangements and margin parameters.

2.1 Notation and Useful Results

Linear Algebra. We write vectors as bold lowercase letters, and denote the \( i \)-th component of a vector \( \mathbf{v} \in \mathbb{R}^N \) by \( v_i \). For a sequence of vectors we often use superscript indices within square brackets, e.g., \( \mathbf{u}^{[1]}, \ldots, \mathbf{u}^{[M]} \), and the \( i \)-th component of the \( j \)-th vector in the sequence is denoted by \( u_{i}^{[j]} \). Matrices are written as bold uppercase and upright letters, and the entry in the \( i \)-th row and \( j \)-th column of a matrix \( \mathbf{A} \in \mathbb{R}^{M \times N} \) is denoted by \( A_{i,j} \). For a vector \( \mathbf{v} \) with \( N \) components, \( \text{diag}(\mathbf{v}) \) denotes the \( (N \times N) \)-matrix with the components of \( \mathbf{v} \) on the main diagonal and zeros elsewhere. As usual, \( \| \mathbf{v} \|_1, \| \mathbf{v} \|_2, \) and \( \| \mathbf{v} \|_\infty \) denote the Manhattan (or \( L_1 \)), the Euclidean (or \( L_2 \)), and the uniform (or \( L_\infty \)) norm of the vector \( \mathbf{v} \), respectively. For a matrix \( \mathbf{A} \), its spectral norm, denoted by \( \| \mathbf{A} \|_2 \), and its Frobenius norm, denoted by \( \| \mathbf{A} \|_F \), are given by

\[
\| \mathbf{A} \|_2 = \max_{\| \mathbf{v} \|_2 = 1} \| \mathbf{A} \mathbf{v} \|_2 \quad \text{and} \quad \| \mathbf{A} \|_2 = \left( \sum_{i,j} A_{i,j}^2 \right)^{1/2} . \tag{2.1}
\]

The following facts are well-known:

\[
\| \mathbf{A} \|_2 \leq \| \mathbf{A} \|_F \quad \text{and} \quad \| \mathbf{A}^T \|_2 = \| \mathbf{A} \|_F = \max_{\| \mathbf{v} \|_2 = 1} \mathbf{v}^T \mathbf{A}^T \mathbf{A} \mathbf{v} . \tag{2.2}
\]
Probability Theory. For a set \( S \), let \( 1_S(x) \in \{0, 1\} \) denote the characteristic function of \( S \) with value \( 1 \) if and only if \( x \in S \). The probability of a measurable set \( S \) is written as \( \Pr[S] \). Given a distribution \( \mathcal{D} \), a random variable \( Z \) distributed according to \( \mathcal{D} \) is denoted by \( Z \sim \mathcal{D} \). The uniform distribution is written as \( U \). For a function \( g \), \( \mathbb{E}_{Z \sim \mathcal{D}}[g(Z)] \) denotes the expected value of \( g(Z) \). If the distribution is clear from context, we often avoid redundancy and omit the subscript \( Z \sim \mathcal{D} \). A probability vector \( q \) is a vector with non-negative components that sum up to \( 1 \). The statistical difference (sometimes called the total variation distance) between two probability distributions \( P, Q \) on a sigma-algebra \( \Sigma \) of subsets of a sample space \( S \) is given by

\[
\text{SD}(P, Q) = \sup_{S \in \Sigma} |P(S) - Q(S)|.
\]

For a finite or countable sample space, the statistical difference can be expressed as

\[
\text{SD}(P, Q) = \frac{1}{2} \|P - Q\|_1 = \frac{1}{2} \sum_{x \in S} |P(x) - Q(x)|.
\]

The following result is well-established.

**Lemma 2.1** (Lehmann and Romano [61]). Let \( x \) be a sample point drawn according to \( P \) or \( Q \) with probability \( 1/2 \), respectively. The task of deciding whether \( x \) was sampled according to \( P \) or \( Q \) has Bayes error \((1 - \text{SD}(P, Q))/2\).

Let \( P, Q \) be two discrete probability distributions over a sample space \( S \). We say that \( P \) is absolutely continuous with respect to \( Q \) (denoted \( P \ll Q \)) if \( Q(x) = 0 \) implies \( P(x) = 0 \) for all \( x \in S \). The Kullback-Leibler divergence from \( Q \) to \( P \) is then defined to be

\[
\text{KLD}(P, Q) = \begin{cases} 
\sum_{x \in S} P(x) \ln \left( \frac{P(x)}{Q(x)} \right) & \text{if } P \ll Q \\
\infty & \text{otherwise}
\end{cases}.
\]

The Kullback-Csizsár-Kemperman inequality (also known as Pinsker’s inequality) bounds the statistical difference between \( P \) and \( Q \) in terms of the Kullback-Leibler divergence from \( Q \) to \( P \) (see for example Devroye [24]).

**Proposition 2.2.** Let \( P, Q \) be two discrete probability distributions over a sample space \( S \). Then

\[
\text{SD}(P, Q) \leq \sqrt{\frac{1}{2} \cdot \text{KLD}(P, Q)}.
\]

In later chapters, we will make use of the following well-known concentration bounds.
2.2 Learning Theory

**Theorem 2.3** (Markov’s inequality). Let $Z$ be a non-negative random variable with finite expected value $\mathbb{E}[Z]$. Then

$$\Pr[Z \geq \xi] \leq \frac{\mathbb{E}[Z]}{\xi},$$

for any $\xi > 0$.

**Theorem 2.4** (Chebyshev’s inequality [22]). Let $Z$ be a random variable with finite expected value $\mathbb{E}[Z]$, and finite variance $\text{Var}[Z] > 0$. Then

$$\Pr[|Z - \mathbb{E}[Z]| \geq \xi] \leq \frac{\text{Var}[Z]}{\xi^2},$$

for any $\xi > 0$.

**Theorem 2.5** (Hoeffding’s inequality [45, 70]). Let $Z_1, \ldots, Z_n$ be independent bounded random variables with $Z_i \in [a, b]$, where $-\infty < a < b < \infty$, for all $i = 1, \ldots, n$. Then

$$\Pr\left[\frac{1}{n} \sum_{i=1}^{n} (Z_i - \mathbb{E}[Z_i]) \geq \xi\right] \leq 2 \exp\left(-\frac{2n\xi^2}{(b - a)^2}\right),$$

for any $\xi \geq 0$.

**Asymptotic Notation.** In this work, we make use of Landau’s big $O$ and big $\Omega$ notations. Let $g$ and $g'$ be two real-valued functions defined on some subset of $\mathbb{R}^N$. We say $g(x) = O(g'(x))$ if and only if there exist positive real numbers $L$ and $C$ such that for all $x$ with $\|x\|_\infty \geq L$, $|g(x)| \leq C|g'(x)|$. We write $g(x) = \Omega(g'(x))$ if and only if $g'(x) = O(g(x))$. Whenever we want to ignore poly-logarithmic factors, we make use of the $\tilde{O}$ notation, which is a variant of Landau’s big $O$ notation. More specifically, $g(x) = \tilde{O}(g'(x))$ if there exists $a > 0$ such that $g(x) = O(g'(x) \log^a g'(x))$.

2.2 Learning Theory

In this section, we restrict our attention to a binary classification problem. Here “learning” consists of approximating a target concept (the labeling procedure which classifies the elements of some universe into two categories) based on a set of observations.

Let $\mathcal{X}$ be a non-empty set, called domain. $\mathcal{X}$ can be thought of as a (possibly infinite) set of features or data records. Moreover, suppose $\ell$ characterizes the representation size of the elements of $\mathcal{X}$, e.g., $\mathcal{X} = \{-1, 1\}^\ell$ or $\mathcal{X} = [0, 1]^\ell$. 

7
Chapter 2. Preliminaries

In the context of learning theory, the elements of $\mathcal{X}$ are usually referred to as instances or examples. A concept over $\mathcal{X}$ is given by a predicate $f: \mathcal{X} \to \{-1, 1\}$. A subset $\mathcal{F} \subseteq \{-1, 1\}^\mathcal{X}$ is called a concept class. We call $\mathcal{H}$ a hypothesis class for $\mathcal{F}$, if $\mathcal{H} \subseteq \{-1, 1\}^\mathcal{X}$ and $\mathcal{F} \subseteq \mathcal{H}$. The elements of $\mathcal{H}$ are called hypotheses.

Valiant’s PAC model of learning [81] considers algorithms that upon receiving examples randomly chosen from some unknown distribution $\mathcal{D}$ over $\mathcal{X}$ and labeled according to an unknown target concept $f^* \in \mathcal{F}$ output a hypothesis $h \in \mathcal{H}$ that approximates $f^*$. More formally, assume there exists an example oracle $\text{EX}(f^*, \mathcal{D})$ which after being invoked returns a pair $(Z, f^*(Z))$, where $Z \overset{i.i.d.}\sim \mathcal{D}$, i.e., independently of any previous example. A concept class $\mathcal{F}$ is (strongly) PAC-learnable if there exists an algorithm $L$ that, for any $f^* \in \mathcal{F}$, any $\alpha, \beta > 0$ and any distribution $\mathcal{D}$ over $\mathcal{X}$, given $\alpha, \beta$ and access to the oracle $\text{EX}(f^*, \mathcal{D})$, outputs, with probability at least $1 - \beta$, a hypothesis $h$ with error at most $\alpha$, i.e., such that $\text{err}_D(h, f^*) := \Pr_{Z \sim \mathcal{D}}[h(Z) \neq f^*(Z)] \leq \alpha$. An exhaustive analysis of the PAC learning model is beyond the scope of this thesis and we refer the interested reader to the books of Kearns and Vazirani [57], and Shalev-Shwartz and Ben-David [76].

2.2.1 Statistical Query Model

The statistical query model (often shortened to SQ-model) of learning was introduced by Kearns [56], and represents a restriction of the PAC learning model. Instead of having access to random examples for producing a hypothesis $h$, in this framework a learning algorithm obtains approximate statistics on the unknown target concept from the $\text{SQ-oracle} \text{STAT}(f^*, \mathcal{D})$.

**Definition 2.1.** A statistical query is a request to $\text{STAT}(f^*, \mathcal{D})$ of the form $(\psi, \tau)$ where $\psi: \mathcal{X} \times \{-1, 1\} \to \{-1, 1\}$ is a function on labeled examples and $\tau \in [0, 1]$ is a tolerance parameter. Given such a query, $\text{STAT}(f^*, \mathcal{D})$ returns a value $v \in \mathbb{R}$ such that

$$\left| \mathbb{E}_{Z \sim \mathcal{D}}[\psi(Z, f^*(Z))] - v \right| \leq \tau .$$

A learning algorithm $L$ that only uses statistical queries to produce a hypothesis $h$ is often called SQ-learner. Following the notation used by Feldman and Kanade [34], a concept class $\mathcal{F}$ is said to be (strongly) learnable from statistical queries or (strongly) SQ-learnable with respect to a distribution $\mathcal{D}$ if there is an SQ-learner $L$ that, for any $f^* \in \mathcal{F}$ and any $\alpha, \beta > 0$, outputs, with probability at least $1 - \beta$, a hypothesis $h$ such that $\text{err}_D(h, f^*) \leq \alpha$. Furthermore, the running time of $L$ must be polynomial in $\ell$, $1/\alpha$ and $1/\beta$ and the statistical queries must be polynomially evaluable and have a tolerance $\tau$ that is lower-bounded by an inverse polynomial in $\ell$ and $1/\alpha$. 
The aforementioned definition captures the computational complexity of learning from statistical queries. On the other hand, the query complexity represents the information-theoretic (or statistical) complexity of SQ-learning. More formally, the query complexity of (strong) SQ-learning $\mathcal{F}$ with respect to a distribution $\mathcal{D}$ and error $\alpha$ is bounded by $q \in \mathbb{N}$ if there exists a (possibly computationally unbounded) SQ-learner $L$ that, for any $f^* \in \mathcal{F}$, makes at most $q$ queries to $\text{STAT}(f^*, \mathcal{D})$, and outputs a hypothesis $h$ such that $\text{err}_{\mathcal{D}}(h, f^*) \leq \alpha$. Moreover, the tolerance $\tau$ must be lower-bounded by an inverse polynomial in $\ell$ and $1/\alpha$.

A class is said to be weakly SQ-learnable with respect to a distribution $\mathcal{D}$ if there exists an SQ-learner $L$ that, for any $f^* \in \mathcal{F}$, produces a hypothesis $h$ such that $\text{err}_{\mathcal{D}}(h, f^*) \leq 1/2 - \alpha$, for some $\alpha > 0$ lower-bounded by an inverse polynomial in $\ell$. The requirement on the running time is the same as in the definition of strong SQ-learnability. Similarly, the query complexity of weak SQ-learning $\mathcal{F}$ is the minimum number of statistical queries required to output a hypothesis with error at most $1/2 - \alpha$. In other words, strong SQ-learnability corresponds to the ability of producing an arbitrarily good SQ-learner for a concept class. On the other hand, for weak SQ-learnability it suffices to output a hypothesis whose error is slightly better than what a random labeling would achieve.

Blum et al. [13] characterize the query complexity of weak SQ-learning a concept class $\mathcal{F}$ (up to a polynomial factor) using a combinatorial parameter called the statistical query dimension (often shortened to SQ-dimension), which corresponds to the maximum number of almost orthogonal concepts in $\mathcal{F}$.

**Definition 2.2** (Blum et al. [13]). For a concept class $\mathcal{F}$ and a distribution $\mathcal{D}$ over $\mathcal{X}$, the statistical query dimension of $\mathcal{F}$ with respect to $\mathcal{D}$, denoted by $\text{SQdim}_D(\mathcal{F})$, is the largest $t \in \mathbb{N}$ such that there exist concepts $f_1, f_2, \ldots, f_t \in \mathcal{F}$ such that for all $1 \leq i < j \leq t$ the following holds:

$$\left| \mathbb{E}_{Z \sim \mathcal{D}}[f_i(Z)f_j(Z)] \right| \leq \frac{1}{t}.$$ 

**Theorem 2.6** (Blum et al. [13]). Let $\mathcal{F}$ be a concept class and $\mathcal{D}$ be a distribution such that $\text{SQdim}_D(\mathcal{F}) = t$. Then the following statements hold.

(i) There exists an SQ-learner for $\mathcal{F}$ with respect to $\mathcal{D}$ that makes $t$ queries, each of tolerance at least $1/(3t^3)$, and outputs a hypothesis with error at most $1/2 - 1/(3t^3)$.

(ii) Any SQ-learner that uses a tolerance parameter lower-bounded by $1/\sqrt{t}$ requires, in the worst case, at least $\sqrt{t}/2$ queries to learn $\mathcal{F}$ with error less than $1/2 - 1/t^3$. 


Chapter 2. Preliminaries

For the sake of completeness, we remark that subsequent works \[87, 79, 35\] greatly simplify the proof and improve the statement of Theorem 2.6. For the purpose of this thesis, however, the original result suffices.

Finally, Feldman and Kanade [34] show that a concept class with polynomial SQ-dimension can be weakly SQ-learned by a polynomial-time algorithm with advice (also referred to as non-uniform algorithm). This implies that the query and computational complexity of weak SQ-learning over a fixed distribution are essentially the same.

### 2.2.2 Linear Arrangements and Margin Parameters

For a finite domain $\mathcal{X}$ and a finite concept class $\mathcal{F}$, it is often convenient to consider the sign matrix $A = (A_{x,f}) \in \{-1, 1\}^{\mathcal{X} \times \mathcal{F}}$ induced by $\mathcal{F}$, i.e., such that $A_{x,f} = f(x)$. As shown by Kallweit and Simon [53], the weak SQ-learnability of $\mathcal{F}$ is closely related to linear arrangements for the sign matrix $A$, and the margins they achieve. In this section, we briefly remind the reader of these relations.

**Definition 2.3.** An $r$-dimensional (homogeneous) linear arrangement for a sign matrix $A \in \mathbb{R}^{M \times N}$ is given by

$$A = \left( u^{[1]}, \ldots, u^{[M]}; v^{[1]}, \ldots, v^{[N]} \right),$$

where $u^{[i]}, v^{[j]}$ are vectors in $\mathbb{R}^r$ of unit Euclidean norm, i.e., $\|u^{[i]}\|_2 = \|v^{[j]}\|_2 = 1$, for $i \in \{1, 2, \ldots, M\}$ and $j \in \{1, 2, \ldots, N\}$.

With an arrangement $\mathcal{A}$ for $A$, we associate the following margin parameters:

$$\gamma_{i,j}(A | \mathcal{A}) = \langle u^{[i]}, v^{[j]} \rangle \cdot A_{i,j},$$

$$\bar{\gamma}_j(A | \mathcal{A}) = \frac{1}{M} \sum_{i=1}^{M} \gamma_{i,j}(A | \mathcal{A}),$$

$$\bar{\gamma}_{\min}(A | \mathcal{A}) = \min_{j=1,\ldots,N} \bar{\gamma}_j(A | \mathcal{A}).$$

Clearly, the value of a margin parameter lies in the interval $[-1, 1]$. The arrangement $\mathcal{A}$ for $A$ is called error-free if none of the margin parameters $\gamma_{i,j}(A | \mathcal{A})$ is negative. The parameter $\bar{\gamma}_{\min}(A | \mathcal{A})$ gives the row-average margin that is guaranteed by $\mathcal{A}$ for every column of $A$. Intuitively, we should think of an arrangement as being “good” if it induces “large margin” parameters. For this reason, we define

$$\bar{\gamma}_{\min}(A) = \max_{\mathcal{A}} \bar{\gamma}_{\min}(A | \mathcal{A}) \quad \text{and} \quad \bar{\gamma}^{ef}_{\min}(A) = \max_{A_{x,f}} \bar{\gamma}_{\min}(A | A_{x,f}),$$

10
where $\mathcal{A}_{ef}$ ranges over error-free arrangements for $\mathcal{A}$ only.

Margin parameters exhibit nice relations to the so-called Forster bound [36]. For the purpose of this thesis, the following variant [53] of the Forster bound is needed:

$$\text{FB}(\mathcal{A}) = \max_q \text{FB}_q(\mathcal{A}) \quad \text{for} \quad \text{FB}_q(\mathcal{A}) = \frac{\sqrt{M}}{\| \mathcal{A} \cdot \text{diag}(q)^{1/2} \|_2}. \quad (2.3)$$

In (2.3), $q$ ranges over all $N$-dimensional probability vectors. As briefly recalled in Section 2.2.1, the number of queries required to weakly SQ-learn $\mathcal{F}$ with respect to the uniform distribution over $\mathcal{X}$ is polynomially related to the SQ-dimension of $\mathcal{F}$ with respect to $\mathcal{U}$. We denote the latter as $\text{SQdim}_{\mathcal{U}}(\mathcal{A})$ (regarding the sign matrix $\mathcal{A}$ as our representation of $\mathcal{F}$). We are now prepared to describe the relations among the various parameters associated with a sign matrix.

**Lemma 2.7** (Kallweit and Simon [53]). Let $\mathcal{A} \in \{-1, 1\}^{M \times N}$ be a sign matrix. Then the following holds:

(i) The parameters $\text{SQdim}_{\mathcal{U}}(\mathcal{A})$, $\text{FB}(\mathcal{A})$, $\gamma_{\min}(\mathcal{A})^{-1}$ are pairwise polynomially related.

(ii) $\gamma_{\min}(\mathcal{A}) = \max_A \min_q \sum_{j=1}^N q_j \gamma_j(\mathcal{A} \mid A)$, where $A$ ranges over all arrangements for $\mathcal{A}$ and $q$ ranges over all $N$-dimensional probability vectors.

The reciprocal value of a margin parameter, like $\gamma_{\min}(\mathcal{A})^{-1}$ in Lemma 2.7, is referred to as margin complexity. The intuition behind it is that a “small margin” corresponds to a “high margin complexity”.
Differential Privacy

In this chapter, we first provide a high-level description of the promise of differential privacy and its meaning. We subsequently introduce the mathematical definition and present some of its properties. We then conclude the chapter analyzing two building block mechanisms which are extensively employed in differential privacy, namely the Laplace and the exponential mechanism. For a more thorough introduction to differential privacy and a survey of the main techniques and tools used, we refer the interested reader to the book of Dwork and Roth [28].

3.1 Motivation and Informal Discussion

The goal of private data analysis is to enable rich and useful statistics on a dataset to be performed while simultaneously providing privacy guarantees to the users whose data are collected and processed. The typical scenario is when a trusted party, e.g., a hospital or a public institution, holds some data, e.g., medical records or voter registration information, and is required to provide publicly available statistical information about it. As some of these data are highly sensitive, there is an obvious need for a mathematically rigorous definition of privacy which is able to address and mitigate the threats that data curation and analysis may elicit. Ideally, we would like the analyst (or information consumer) to learn useful information about a population while preventing any disclosure about an individual.

In recent years, differential privacy [30] has emerged as an answer to this problem and is currently regarded to as the leading paradigm for privacy-preserving data analysis. Differential privacy formally ensures that aggregate statistics output by a randomized algorithm are not significantly influenced by the presence or absence of an individual datum. In particular, this guarantees
that the same result is achieved independently of whether any individual refuses
to be part of the dataset or not, thereby protecting her privacy.

Privacy-preserving data analysis has been an extensively studied topic for
many years, long before the introduction of differential privacy. Previous ap-
proaches though were not completely satisfactory and showed some flaws. For
instance, anonymization is a simplistic answer to the problem, where portions
of the data records are concealed and the rest is made public for analysis. Nev-
ertheless, the granularity of the data themselves and the abundance of publicly
available and non-anonymized data sources can make “de-anonymization” in-
credibly simple. This threat may result in so-called linkage attacks, and the
success of the reconstruction capabilities has been demonstrated in a number
of astonishing works. For example, Sweeney \[78\] carried out a linkage attack
against anonymized medical encounter data and Narayanan and Shmatikov \[68\]
linked the Netflix Prize dataset, which consists of anonymized movie ratings of
Netflix subscribers, to the Internet Movie Database (IMDb). Notwithstanding,
differential privacy is robust against this kind of threats. In fact, differential
privacy is defined as a property of the mechanism itself and is independent
of the side knowledge that an adversary may have. Therefore, any auxiliary
information available to the adversary does not increase her capability of re-
constructing an unknown individual record in the dataset.

Albeit there is by now an extensive literature in differential privacy spanning
multiple disciplines and significant achievements have been made, a consider-
able number of problems remains open. This work tackles some of these chal-
lo\[2\]lenges, specifically regarding the design of generic differentially private mecha-
nisms and the privacy-utility trade-off that naturally arises.

### 3.2 Definition and Properties

As in Section 2.2 let \( \mathcal{X} \) be an arbitrary (possibly infinite) domain, whose
elements are characterized by a representation size \( \ell \). Given \( n \in \mathbb{N} \), a database \( d \)
(sometimes referred to as dataset) is a vector of \( n \) entries from \( \mathcal{X} \), i.e., \( d \in \mathcal{X}^n \).

We refer to \( n \) as the size of the database \( d \). Two databases \( d, d' \in \mathcal{X}^n \) are said
to be neighboring (denoted \( d \approx d' \)) if they differ in one entry only, i.e., there
exists exactly an index \( i \in \{1, 2, \ldots, n\} \) such that \( d_i \neq d'_i \). In other words, \( d \)
and \( d' \) can be obtained from each other by a single record exchange.

Let \( \mathcal{X}^* = \bigcup_{n>0} \mathcal{X}^n \). A mechanism \( \mathcal{M} : \mathcal{X}^* \rightarrow \mathcal{R} \) is a randomized algorithm
(meaning that, for every \( d \in \mathcal{X}^* \), \( \mathcal{M}(d) \) is an \( \mathcal{R} \)-valued random variable) that

\[^2\text{Representing databases as vectors is very common in the differential privacy literature. In this thesis, we will also present and make use of other representations, e.g., histograms, whenever this helps simplify our analysis.} \]
takes as input a database over a domain $X$ and produces an output in some response universe $R$.

**Definition 3.1** ([30, 29]). Let $R$ be a (possibly infinite) set of responses. A mechanism $M : X^n \to R$ is said to provide $(\varepsilon, \delta)$-differential privacy for $\varepsilon > 0$ and $0 \leq \delta < 1$ if, for every $n \in \mathbb{N}$, every pair $(d, d') \in X^n \times X^n$ of neighboring databases, and every measurable $S \subseteq R$, the condition

$$\Pr [M(d) \in S] \leq e^\varepsilon \Pr [M(d') \in S] + \delta$$

(3.1)

holds. Here, the probability space is over the internal randomization of the mechanism $M$. If $\delta = 0$ we simply say that $M$ provides $\varepsilon$-differential privacy.

In order to provide differential privacy, a mechanism $M$ must limit the influence of input data on the induced response distribution. By doing so, a powerful adversary with knowledge of all but one input record, the mechanism up to random source, and possibly unbounded computation power can only gain very limited (dependent on the privacy parameter $\varepsilon$) information on an unknown input datum by observing the mechanism responses.

In differential privacy, it is commonly assumed that there exists a trustworthy party that has access to the input database $d$ and is responsible for transforming $d$ into $M(d)$. We make such assumption in Chapters 5 and 6. Definition 3.1 also applies to scenarios where we cannot rely on such entity. This setting will be discussed in detail in Chapter 7.

We underline that $(\varepsilon, \delta)$-differential privacy provides weaker privacy guarantees than $\varepsilon$-differential privacy. Specifically, $(\varepsilon, \delta)$-differential privacy only ensures that it is unlikely (and the confidence is controlled by $\delta$) that the induced response distribution significantly differs on pairs of neighboring databases. In other words, $(\varepsilon, \delta)$-differential privacy guarantees that, for all neighboring databases $d, d'$,

$$\left| \ln \left( \frac{\Pr [M(d) \in S]}{\Pr [M(d') \in S]} \right) \right| \leq \varepsilon$$

with probability at least $1 - \delta$, where the probability space is over the internal randomization of the mechanism $M$. This formulation is not immediate from Definition 3.1, but was proved by Dwork et al. [32]. In contrast, when $\delta = 0$, condition (3.1) holds for every pair of neighboring databases and every response. In this work, we will mostly focus on $\varepsilon$-differentially private mechanisms, as there generally is a significant motivation for achieving stronger privacy guarantees.

In addition to the sensitive database $d$, the mechanism $M$ often takes some auxiliary parameters as inputs. For instance, we may specify a vector-valued query $g : X^n \to \mathbb{R}^r$ on the database $d$ and the privacy parameter $\varepsilon$. The
mechanism $M_{g,\varepsilon}(d)$ would then output an $\varepsilon$-differentially private approximation of $g(d)$. If such additional parameters are clear from context, we keep our notation as simple as possible and omit them.

Differential privacy partly owes its popularity to some beneficial properties it exhibits which made its adoption very manageable. Here, we briefly present the properties we will often use throughout this thesis. The corresponding simple proofs can be found in the survey of Dwork and Roth [28].

**Proposition 3.1 (Group privacy [30, 28])**. Let $M: \mathcal{X}^* \rightarrow \mathcal{R}$ be an $(\varepsilon, \delta)$-differentially private mechanism. Then, for every pair $(d, d') \in \mathcal{X}^n \times \mathcal{X}^n$ of databases that differ in $k$ entries, and every measurable $S \subseteq \mathcal{R}$, the following holds:

$$\Pr[M(d) \in S] \leq e^{k\varepsilon} \Pr[M(d') \in S] + ke^{(k-1)\varepsilon}\delta.$$  \hfill (3.2)

**Proposition 3.2 (Post-processing [30])**. Let $M: \mathcal{X}^* \rightarrow \mathcal{R}$ be an $(\varepsilon, \delta)$-differentially private mechanism. Let $M': \mathcal{R} \rightarrow \mathcal{R}'$ be an arbitrary random mapping. Then $M' \circ M: \mathcal{X}^* \rightarrow \mathcal{R}'$ provides $(\varepsilon, \delta)$-differential privacy.

**Proposition 3.3 (Sequential composition [29, 27])**. Let $M_i: \mathcal{X}^* \rightarrow \mathcal{R}_i$ be an $(\varepsilon_i, \delta_i)$-differentially private mechanism, for $i = 1, \ldots, r$. Then the mapping

$$M(d): \mathcal{X}^* \rightarrow \prod_{i=1}^{r} \mathcal{R}_i$$

such that $M(d) = (M_1(d), \ldots, M_r(d))$ provides $(\sum_{i=1}^{r} \varepsilon_i, \sum_{i=1}^{r} \delta_i)$-differential privacy.

Proposition 3.1 is an immediate consequence of Definition 3.1. As the number of entries in which the databases differ gets larger, the constraint on the induced response distribution becomes less restrictive. Proposition 3.2 states that, without any additional information on the sensitive database, an adversary cannot compromise privacy by composing the mechanism’s output with a data-independent mapping. Finally, Proposition 3.3 shows that the composition of differentially private mechanisms provides differential privacy, up to a linear degradation of the privacy parameters.

### 3.3 Basic Tools

In the previous section we introduced the concept of differentially private mechanisms but we never mentioned how to actually construct such algorithms. In this section, we present two of the most renowned mechanisms used to achieve differential privacy, namely the Laplace [30] and the exponential mechanism [66]. These are widely used tools that permit researchers and practitioners to effectively implement privacy-preserving mechanisms. Furthermore,
these simple algorithms are extensively employed as building blocks in the design of more involved mechanisms, capable of providing privacy for complex analytical tasks.

3.3. Basic Tools

The following distribution has proved remarkably useful in differential privacy.

**Definition 3.2.** For any \( \mu \in \mathbb{R} \) and \( \lambda > 0 \), the Laplace distribution \( \text{Lap}(\mu, \lambda) \) is the distribution over the reals given by the following probability density function:

\[
L_D(x) = \frac{1}{2\lambda} \exp\left(-\frac{|y - \mu|}{\lambda}\right).
\]

When \( \mu = 0 \), we simply write \( \text{Lap}(\lambda) \). The \( r \)-dimensional Laplace distribution is the product of \( r \) independent (1-dimensional) Laplace distributions \( \text{Lap}(\mu_i, \lambda_i) \), for \( i = 1, \ldots, r \). It is denoted by \( \text{Lap}((\mu_1, \ldots, \mu_r), (\lambda_1, \ldots, \lambda_r)) \). As for the 1-dimensional case, if \( \mu = 0 \) we simply write \( \text{Lap}(\lambda) \). Finally, if for some \( \lambda > 0 \), \( \lambda_1 = \ldots = \lambda_r = \lambda \), we further simplify our notation and write \( \text{Lap}(\lambda)^r \) instead of \( \text{Lap}((\lambda, \ldots, \lambda)) \).

The Laplace mechanism [30] is a very simple tool for privatizing vector-valued functions: adding zero-mean Laplace noise provides privacy if the noise is calibrated to the function’s sensitivity.

**Definition 3.3 (Dwork et al. [30]).** The sensitivity of a function \( g : \mathcal{X}^n \to \mathbb{R}^r \) is given by

\[
\Delta(g) = \sup_{d \approx d'} \|g(d) - g(d')\|_1,
\]

where the supremum is taken over all \( d, d' \in \mathcal{X}^n \) that differ in one entry only.

The sensitivity of a function \( G : \mathcal{X}^n \times \mathcal{Y} \to \mathbb{R}^r \) is defined as

\[
\Delta(G) = \sup_{y \in \mathcal{Y}} \Delta(G(\cdot, y)).
\]

**Lemma 3.4 (Dwork et al. [30]).** Let \( g : \mathcal{X}^n \to \mathbb{R}^r \) be a function of finite sensitivity, \( d \in \mathcal{X}^n \), and \( \varepsilon > 0 \). Then, the Laplace mechanism

\[
\mathcal{M}_{\text{Lap}}^g(d) \sim \text{Lap}(g(d), \lambda)
\]

provides \( \varepsilon \)-differential privacy if \( \lambda_i = \Delta(g)/\varepsilon \) for every \( i \in \{1, 2, \ldots, r\} \).

Note that the Laplace mechanism does not necessarily require the function \( g \) as input in order to provide differential privacy. In fact, it suffices to have oracle access to the target function evaluation \( g(d) \) and the sensitivity \( \Delta(g) \).
3.3.2 Exponential Mechanism

The exponential mechanism was introduced by McSherry and Talwar [66] for coping with situations in which adding noise directly to the output may torpedo the utility of the computed quantity, or when evaluating algorithms with arbitrary (and possibly non-numeric) response universes $\mathcal{R}$ while providing differential privacy.

The exponential mechanism is defined with respect to a non-private utility (or score) function $U : \mathcal{X}^n \times \mathcal{R} \rightarrow \mathbb{R}$, which benchmarks the “quality” of the output responses on a given database. Given a sensitive database $d \in \mathcal{X}^n$, the exponential mechanism is designed so as to output a response $S \subseteq \mathcal{R}$ with probability proportional to its score function. More specifically, the probability associated with $S$ increases exponentially with its utility on $d$. This has the effect to bias the distribution towards responses with higher utilities, thereby increasing the expected score.

**Definition 3.4** (McSherry and Talwar [66]). Let $\mathcal{R}$ be an arbitrary response universe. Let furthermore $U : \mathcal{X}^n \times \mathcal{R} \rightarrow \mathbb{R}$ be a utility function of finite sensitivity $\Delta(U)$ (as per Definition 3.3), $d \in \mathcal{X}^n$, and $\varepsilon > 0$. The exponential mechanism $M_{\mathcal{R},U,\varepsilon}(d)$ assigns to every $s \in \mathcal{R}$ a probability density proportional to

$$
\exp\left(\frac{\varepsilon \cdot U(d,s)}{\Delta(U)}\right).
$$

(3.3)

It then returns a value sampled from such distribution.

**Lemma 3.5** (McSherry and Talwar [66]). The exponential mechanism provides $2\varepsilon$-differential privacy.

We briefly note that the original definition of McSherry and Talwar [66] is slightly more general and has an additional factor $\mu(s)$ in (3.3), which represents a prior distribution on $\mathcal{R}$. In this thesis, we only deal with uniform priors and have therefore omitted $\mu(s)$ from Definition 3.4.

In several cases (see for example the unit demand auction setting discussed by McSherry and Talwar [66]) the factor 2 in the statement of Lemma 3.5 can be removed, strengthening the privacy guarantees to $\varepsilon$-differential privacy.

We conclude this section by observing that the exponential mechanism defines a distribution over a possibly large response universe $\mathcal{R}$. Therefore, the exponential mechanism may not be efficiently implementable in general. For instance, this might be unavoidable when $\mathcal{R}$ is super-polynomially large in the relevant parameters of the problem.
A Lower Bound on the Release of Integer Partitions

We consider the problem of releasing integer partitions under differential privacy. Such a problem is of high relevance for many real-world applications, from the publication of password frequency lists to the release of the degree distribution of social networks. In this chapter, we provide a lower bound on the minimax risk that any differentially private mechanism releasing an integer partition must incur.

This chapter is based on the paper “A Lower Bound on the Release of Differentially Private Integer Partitions” [5], which is a joint work of the author with Hans Ulrich Simon. All results in this chapter originally appeared in this paper.

4.1 Introduction

Since its introduction, differential privacy has seen a number of different applications. A recent one deals with the private release of integer partitions [12]. Throughout this chapter, whenever we write that a mechanism releases a partition of a non-negative integer \( n \) we always assume that the input is a partition of \( n \) but the output is allowed to be a partition of any non-negative integer. In their work, Blocki et al. [12] show that the exponential mechanism of McSherry and Talwar [66] (see Section 3.3.2 for further details) can be used to release the partition of a non-negative integer \( n \) with error \( O(\sqrt{n}/\varepsilon) \), where the error is measured in terms of the \( L_1 \)-distance between the input and output partition. Blocki et al. [12] also propose an approximate instantiation of the exponential mechanism which attains the same error bound but, in contrast to the latter, achieves computational efficiency by relaxing the privacy guarantees provided to \( (\varepsilon, \delta) \)-differential privacy.
The release of integer partitions is of high practical relevance. For instance, Blocki et al. [12] use this algorithm to publish a password frequency list from a password dataset of 70 million Yahoo! users [15]. In fact, a password frequency list is the partition of the number of passwords in a dataset. The security community is highly interested in such lists since they enable a better understanding of how passwords are chosen by users. Moreover, they can be used to accurately estimate security risks and design new password defenses. Since password frequency lists obviously contain highly sensitive information, users’ privacy needs to be appropriately protected. As observed by Blocki [11], integer partitions have other applications, besides password frequency lists. The degree distribution of a graph $G$ with $K$ vertices and $E$ edges is a partition of the integer $2E$, and has been studied under various privacy models [44, 55, 72].

Besides their upper bound on utility, Blocki et al. [12] give empirical evidence that the error of their mechanism seems to scale with $1/\sqrt{\epsilon}$ instead of $1/\epsilon$ for large values of $n$. In this chapter, we demonstrate that this is actually the best accuracy we can hope for, proving that any $\epsilon$-differentially private mechanism which publishes a partition of the integer $n$ must incur a minimax risk of order $\Omega(\sqrt{n/\epsilon})$ if $n \geq 1/(2\epsilon)$. Moreover, we show that the lower bound becomes $\Omega(n)$ if $n < 1/(2\epsilon)$. Since an $\epsilon$-differentially private mechanism which always returns the partition of 0 incurs an $L_1$-error of at most $n$, our lower bound for small values of $n$ is optimal.\(^3\) These results are particularly interesting in view of Blocki’s conjecture [11] that for sufficiently large values of $n$ the upper bound of Blocki et al. [12] could actually be improved to $O(\sqrt{n/\epsilon})$. If this conjecture held, then our lower bound would actually be tight in this regime, too.

Our proof is based on an application of Assouad’s lemma [7], a well-known tool for establishing lower bounds in statistics which has proved very useful in other differential privacy applications [41, 26]. Using this technique, we are able to improve a previous result due to Blocki [11], who showed a weaker lower bound of order $\Omega(\sqrt{n}/\log n)$.

\[ 4.2 \quad \text{Setting} \]

We remind the reader that a database $d$ is a collection of records from a (possibly infinite) universe $\mathcal{X} = \{x_1, x_2, \ldots\}$. In general $\mathcal{X}$ could be uncountable, but in this chapter it suffices to restrict our attention to countable universes only.

\(^3\)In this chapter, when we write optimal we always mean optimal modulo a constant factor.
Databases as Histograms. Throughout this chapter, we will conveniently represent databases by their histograms, i.e., a database $d$ over $\mathcal{X}$ is viewed as a sequence $(n_1, n_2, \ldots) \in \mathbb{N}^\mathcal{X}$ where $n_i$ represents the number of elements of type $x_i$. Note that a database of size $n$ is a sequence $d$ of non-negative integers which sum up to $n$, and that at most $n$ members of the sequence $d$ are non-zero. Hereafter, we also identify a finite sequence $n_1, \ldots, n_k$ of non-negative integers with the infinite sequence $n_1, \ldots, n_k, 0, 0, \ldots$. Hence, databases are always given by infinite sequences of non-negative integers (even when the underlying universe $\mathcal{X}$ is finite). The $L_1$-distance between two databases $d, d'$ is thus $\|d - d'\|_1 = \sum_{i \geq 1} |n_i - n'_i|$. In contrast to Section 3.2, in this chapter we say that two databases are neighboring if and only if their $L_1$-distance equals 1. We recall that, according to Proposition 3.1, an $\varepsilon$-differentially private mechanism $\mathcal{M}$ provides $k\varepsilon$-differential privacy when it is applied to databases $d, d'$ such that $\|d - d'\|_1 \leq k$.

Integer Partitions as Sorted Histograms. We define a partition of an integer $n$ as a non-increasing sequence $\rho = (\rho_1 \geq \rho_2 \geq \ldots \geq \rho_n)$ such that $\rho_1, \rho_2, \ldots, \rho_n$ are non-negative integers that sum up to $n$. We will often identify a partition $\rho$ of the integer $n$ by an infinite sequence by setting $\rho_k = 0$ for all $k > n$. If $d = (n_i)_{i \geq 1}$ is a database of size $n$ and, for $k = 1, \ldots, n$, $\rho_k$ is the $k$-th largest member of the sequence $d$, we say that $\rho(d) = (\rho_1, \ldots, \rho_n)$ is the partition of the integer $n$ that is induced by $d$. Note that the integer partition induced by $d$ provides less information than $d$ itself because it hides which member of the universe actually occurs most frequently (or second-most frequently and so on). In the sequel, $\mathcal{P}_n$ denotes the set of all partitions of the integer $n$ and $\mathcal{P} = \bigcup_{n \geq 0} \mathcal{P}_n$.

Let $d, d'$ be two databases and let $\rho(d), \rho(d')$ denote the corresponding integer partitions. The $L_1$-distance between $d$ and $d'$ is, in general, not the same as the $L_1$-distance between $\rho(d)$ and $\rho(d')$. For instance, the $L_1$-distance between $d = (1, 2)$ and $d' = (2, 1)$ is 2 but the corresponding integer partitions coincide, i.e., $\rho(d) = \rho(d') = (2, 1)$. The following result is rather straightforward.

Proposition 4.1. The following holds.

(i) Let $\rho(d)$ and $\rho(d')$ be the integer partitions that are induced by the databases $d$ and $d'$, respectively. Then $\|\rho(d) - \rho(d')\|_1 \leq \|d - d'\|_1$.

(ii) For every pair of integer partitions $\rho, \rho'$, there exist databases $d$ and $d'$ such that $d$ induces $\rho$, $d'$ induces $\rho'$, and $\|\rho - \rho'\|_1 = \|d - d'\|_1$.

In this chapter, we consider $\varepsilon$-differentially private mechanisms $\mathcal{M}_{n,\varepsilon} : \mathbb{N}^\mathcal{X} \rightarrow \mathcal{P}$ that take as input a database of size $n$ and output an integer partition
Chapter 4. A Lower Bound on the Release of Integer Partitions

As observed before, each database \( d \) of size \( n \) induces an integer partition \( \rho(d) \in \mathcal{P}_n \). We can thus define the following risk function:

\[
R(d, \mathcal{M}_{n,\varepsilon}) = \sum_{\hat{\rho} \in \mathcal{P}} \| \hat{\rho} - \rho(d) \|_1 \Pr[\mathcal{M}_{n,\varepsilon}(d) = \hat{\rho}] .
\]

The minimax risk is then given by

\[
R^* = \inf_{\mathcal{M}_{n,\varepsilon}} \left( \sup_{d \in \mathbb{N}^n : \|d\|_1 = n} R(d, \mathcal{M}_{n,\varepsilon}) \right) .
\]

4.3 Lower Bound Based on Assouad’s Lemma

Two bit vectors \( b, b' \in \{0, 1\}^m \) are said to be \( j \)-neighbors, denoted \( b \approx_j b' \), if they differ in their \( j \)-th bit and coincide on all other bits. Moreover, we write \( b \approx b' \) if \( b \approx_j b' \) holds for some \( j \in \{1, 2, \ldots, m\} \).

Lemma 4.2 (Assouad’s lemma \[7, 88\]). Let \( \{P_b : b \in \{0, 1\}^m\} \) be a family of probability distributions over a discrete domain and let \( V \) denote a set of parameter vectors. For each \( b \in \{0, 1\}^m \), let \( \theta[b] \in V \) denote a parameter vector associated with \( P_b \). Let \( \hat{\theta}[b] = \hat{\theta}[b](X) \) be an estimator of \( \theta[b] \) based on \( X \sim P_b \). For each \( j \in \{1, 2, \ldots, m\} \), let \( \phi_j \) be a pseudo-metric on \( V \). Suppose that the mapping \( \phi : V \times V \to \mathbb{R}_0^+ \) satisfies

\[
\phi(x, y) \geq \sum_{j=1}^m \phi_j(x, y) .
\]

Let \( w \geq 0 \) be chosen such that

\[
\phi_j(\theta[b], \theta[b']) \geq w
\]

holds for every \( j \in \{1, 2, \ldots, m\} \) and for every pair \( b \approx_j b' \) of \( j \)-neighbors in \( \{0, 1\}^m \). Under these assumptions, the following holds:

\[
\max_{b \in \{0, 1\}^m} \mathbb{E}_{X \sim P_b} \left[ \phi(\hat{\theta}[b], \hat{\theta}[b']) \right] \geq \frac{mw}{2} \left( 1 - \frac{1}{2} \max_{b = b'} \|P_b - P_{b'}\|_1 \right) . \tag{4.1}
\]

The credit for the following result should mainly be given to Hall et al. \[41\]. In fact, even though the assertion of Corollary 4.3 is not explicitly mentioned by Hall et al. \[41\], it is sort of immediate from the proof of Proposition 3.1 in their paper.
Corollary 4.3. For every $b \in \{0,1\}^m$, let $d^{[b]}$ be a database. Let $\mathcal{M}$ be an $\varepsilon$-differentially private mechanism that, when applied to $d^{[b]}$, returns a noisy version $\hat{\theta}^{[b]}$ of the true response $\theta^{[b]}$. Let $P_b$ be the distribution on the responses of $\mathcal{M}$ when it is applied to $d^{[b]}$. Suppose that $V, (\phi_j)_{j \in \{1,2,\ldots,m\}}, \phi$ and $w$ have the same meaning and satisfy the same assumptions as in Assouad’s lemma. Finally, suppose that, for every choice of $b \approx b'$, $\|d^{[b]} - d^{[b']}\|_1 \leq \frac{1}{2\varepsilon}$.

Under these assumptions, the following holds:

$$\max_{b \in \{0,1\}^m} \mathbb{E}_{\theta^{[b]} \sim P_b} \left[ \phi \left( \theta^{[b]}, \hat{\theta}^{[b]} \right) \right] \geq \frac{mw}{4}.$$  \(4.2\)

Proof. The assumptions in the corollary imply that, for every possible response $\hat{\theta}$ of $\mathcal{M}$, we have

$$\Pr \left[ \mathcal{M} \left( d^{[b]} \right) = \hat{\theta} \right] / \Pr \left[ \mathcal{M} \left( d^{[b']} \right) = \hat{\theta} \right] \leq e^{1/2}.$$

It follows that $\text{KLD}(P_b, P_{b'}) \leq 1/2$. An application of Proposition 2.2 yields $\|P_b - P_{b'}\|_1 \leq \sqrt{2} \cdot (1/2) = 1$. The corollary is now immediate from Lemma 4.2.

We now apply Corollary 4.3 with the parameters $V, \theta^{[b]}$ and $\hat{\theta}^{[b]}$ replaced by the set $\mathcal{P}$ of integer partitions, by the integer partition $\rho^{[b]}$ induced by a database $d^{[b]}$, and by the noisy output $\hat{\rho}^{[b]}$, respectively.

Theorem 4.4. Let $\varepsilon \leq 1/8$. Then, any $\varepsilon$-differentially private mechanism $\mathcal{M}_{n,\varepsilon}$ releasing an integer partition in $\mathcal{P}$ must incur a minimax risk

$$R^* \geq \begin{cases} \frac{1}{4} \sqrt{\frac{n}{2\varepsilon}} \cdot (1 - o(1)) & \text{if } n \geq \frac{1}{2\varepsilon} \\ \frac{n - 1}{4} & \text{if } n < \frac{1}{2\varepsilon} \end{cases}.$$  \(4.3\)

Proof. We analyze the two cases separately. Assume first that $n \geq 1/(2\varepsilon)$. For two integer partitions $\rho, \rho'$ and $j \geq 1$, let $\phi_j$ be the pseudo-metric given by $\phi_j(\rho, \rho') = |\rho_j - \rho'_j|$. We set $\phi(\rho, \rho') = \sum_{j=1}^{m} \phi_j(\rho, \rho')$ where $m = m(n, \varepsilon)$ is a non-negative integer that will be specified below. Our strategy is to apply Corollary 4.3. To this end, we will specify integer partitions $(\rho^{[b]}_{b \in \{0,1\}^m})$ of $n$ induced by suitably chosen databases $d^{[b]}$. We may view $\rho^{[b]}$ as the true answer when the database is $d^{[b]}$. The mechanism $\mathcal{M}_{n,\varepsilon}$, when applied to $d^{[b]}$, will return a noisy version $\hat{\rho}^{[b]}$ of $\rho^{[b]}$. As in Corollary 4.3, we denote by $P_b$ the
distribution on the responses of $\mathcal{M}_{n,\varepsilon}$ when $\mathcal{M}_{n,\varepsilon}$ is applied to $d^{[b]}$. Let $w = \lfloor 1/(4\varepsilon) \rfloor$. For $k = 1, \ldots, m$, we define

$$\rho_k^{[b]} = 1 + (m - k)w + b_k w .$$

It is evident that, regardless of how $b \in \{0, 1\}^m$ is chosen, the sequence $\rho^{[b]}$ is non-increasing so that it forms an integer partition, indeed. In order to make $\rho^{[b]}$ an integer partition of the number $n$ (for every choice of $b \in \{0, 1\}^m$ including the all-ones vector), we have to make sure that

$$\sum_{k=1}^{\lfloor n/(w+1) \rfloor} (1 + (m - k)w) = m + \frac{1}{2}wm(m + 1) \leq n , \quad (4.4)$$

and we have to extend the sequence $(\rho_k^{[b]})_{k=1,\ldots,m}$ by a sequence of $1$’s followed by a sequence of $0$’s such that $\sum_{k=1}^{\lfloor n/w \rfloor} \rho_k^{[b]} = n$. Note that the extended sequence $\rho^{[b]}$ is still non-increasing. Choosing

$$m = \left\lfloor \frac{2n}{w} \right\rfloor - 1 , \quad (4.5)$$

condition (4.4) certainly holds. Note that our assumption $n \geq 1/(2\varepsilon)$ and the choice $w = \lfloor 1/(4\varepsilon) \rfloor$ imply $m \geq 1$. Suppose that $b \approx_{j} b'$. Then, obviously, we have that $\phi(\rho^{[b]}, \rho^{[b']}) = \phi_j(\rho^{[b]}, \rho^{[b']}) = w$. The $L_1$-distance between the non-extended versions $\rho^{[b]}$ and $\rho^{[b']}$ (with only the first $m$ members of these sequences) yields the same value, $w$. But, since we have extended these sequences (so as to become partitions of the number $n$), the $L_1$-distance doubles, i.e., it equals $2w$. If the databases $d^{[b]}$ and $d^{[b']}$ are properly chosen, their $L_1$-distance is $2w$ as well (see Proposition 4.1). Since $2w \leq 1/(2\varepsilon)$, the assumptions of Corollary 4.3 are satisfied. We conclude that (4.2) is valid. Since $w = \lfloor 1/(4\varepsilon) \rfloor$ and $m$ is chosen according to (4.5), the lower bound $mw/4$ from Corollary 4.3 equals asymptotically the upper term of the right hand-side of (4.3). Since $\phi(\cdot, \cdot)$ lower bounds the $L_1$-metric, the first lower bound is obtained.

Finally, assume $n < 1/(2\varepsilon)$. We are going to exploit Corollary 4.3 once more. For two integer partitions $\rho, \rho'$ and $j = 1$, the pseudo-metric we now consider is given by $\phi(\rho, \rho') = \phi_1(\rho, \rho') = \|\rho - \rho'\|_1$. Let $\rho^{[0]} = (n, 0, \ldots, 0)$ and $\rho^{[1]} = ([n/2] + \sigma, [n/2], 0, \ldots, 0)$ be partitions of $n$, where

$$\sigma = \begin{cases} 1 & \text{if } n \text{ is odd} \\ 0 & \text{otherwise} \end{cases} .$$

Note that $\phi(\rho^{[0]}, \rho^{[1]}) = n - \sigma$. By Proposition 4.1, there exists a pair of databases $d^{[0]}$ and $d^{[1]}$ which induce $\rho^{[0]}$ and $\rho^{[1]}$, respectively. Moreover, $d^{[0]}$
4.4. Related Work

and \(d^{[1]}\) satisfy \(\|d^{[0]} - d^{[1]}\|_1 = n - \sigma\). Choosing \(w = n - \sigma\), the assumptions of Corollary 4.3 are satisfied. In particular, note that our assumption on \(n\) implies \(w \leq 1/(2\varepsilon)\). We then obtain \(\mathfrak{R}^* \geq mw/4 = (n - \sigma)/4 \geq (n - 1)/4\), concluding the proof of Theorem 4.4.

4.4 Related Work

Standard techniques for proving lower bounds in differential privacy, e.g., packing [43] or information-theoretic [65] arguments, cannot be fully leveraged when dealing with integer partitions. The packing technique introduced by Hardt and Talwar [43] (suitably adapted so as to use the \(L_1\)-error instead of the \(L_2\)-error) leads to the lower bound \(\Omega(1/\varepsilon)\) on the \(L_1\)-error of any \(\varepsilon\)-differentially private mechanism that takes a sequence of reals as input and returns a noisy version of the corresponding sorted sequence. This problem is related to the problem of releasing an integer partition but it is not quite the same. The two problems get closer when we replace real sequences by sequences of non-negative integers. For this purpose, we may employ the discretization method discussed by Hardt and Talwar [43]. But then we run into the following dilemma. In order to control the inaccuracy induced by discretization, one has to use non-negative integers within the sequence of length \(n\) which are bigger than \(n/\varepsilon\). These numbers cannot sum up to \(n\) which would be required for applying this technique to integer partitions. Therefore, such technique does not seem to be appropriate for the special sequences that represent integer partitions.

The method based on mutual information [65] can be applied to the problem of releasing integer partitions but the bound obtained is inferior to what a direct application of Assouad’s lemma yields.

4.5 Conclusions and Open Problems

In this chapter we considered the problem of privately releasing integer partitions. In particular, by an application of Assouad’s lemma [7] we showed that any \(\varepsilon\)-differentially private mechanism \(\mathcal{M}_{n,\varepsilon}\) releasing an integer partition of \(n\) must incur a minimax risk \(\Omega(\sqrt{n/\varepsilon})\) if \(n \geq 1/(2\varepsilon)\). We also provided an optimal lower bound of order \(\Omega(n)\) if \(n < 1/(2\varepsilon)\). Although there is empirical evidence suggesting that the exponential mechanism achieves error bound \(O(\sqrt{n/\varepsilon})\) for sufficiently large values of \(n\) [12] [11], proving a matching upper bound and thus the optimality of this mechanism remains an interesting open problem. We shed some light on this question in Chapter 5 where we investigate under which conditions the exponential mechanism is actually optimal in terms of the average-case error introduced. Another compelling open question
is to provide lower bounds on the release of integer partitions under weaker privacy definitions, \( e.g., (\epsilon, \delta) \)-differential privacy when \( \delta > 0 \).
Chapter 5

Average-case Optimality of the Exponential Mechanism

In this chapter, we more thoroughly investigate the exponential mechanism presented in Section 3.3.2. In particular, we study the optimality of the error introduced by the exponential mechanism in the average-case scenario, when the input/output universe of the mechanism can be modeled as a graph where each node is associated with a database. By leveraging linear programming theory, we provide some regularity conditions on the graph structure under which the exponential mechanism is optimal. Moreover, we give a toy example in which the optimality is preserved (up to a constant factor) even if these regularity conditions hold only to a certain extent.

This chapter is based on the paper “On the Optimality of the Exponential Mechanism” [4], which is a joint work of the author with Hans Ulrich Simon. All results in this chapter first appeared in this paper.

5.1 Introduction

In this chapter we restrict our attention to the exponential mechanism introduced by McSherry and Talwar [66]. As discussed in Section 3.3.2, the exponential mechanism is one of the most widely used tools in differential privacy, especially in applications where the response set is arbitrary and possibly non-numeric.

In this chapter we aim to investigate under which conditions the exponential mechanism is optimal in terms of the average-case error. More specifically, we consider the setting where the input and output universe of a privacy mechanism coincide and can be modeled as a graph, where each node is associated with a database, and adjacent nodes correspond to neighboring databases. The optimal privacy mechanism can then be expressed as the solution of a
linear program, where we seek to minimize the average error introduced by the mechanism subject to the constraints induced by differential privacy (see Definition 3.1). We show that, if the induced graph has a transitive automorphism group and a so-called regular layer sequence (whose definition will be provided in Section 5.2), then the exponential mechanism is actually optimal, i.e., its solution coincides with that of the optimal privacy mechanism. We then provide a toy example in which this result holds (up to a constant factor) even if the aforementioned conditions are met only to a large extent.

5.2 Optimal Mechanisms and Linear Programming

Let $G = (K, E)$ denote a graph with $K = |K|$ nodes, $E = |E|$ edges, and diameter $D$. Intuitively, we should think of each node $x \in K$ as a piece of information associated with a database $d \in X^n$. Moreover, we should think of adjacent nodes in $G$ as nodes whose underlying databases are neighboring, i.e., they can be obtained from each other by a single record exchange (see Section 3.2). Hence a node $y$ has distance $\eta$ from another node $x$ if and only if $\eta$ is the smallest number of record exchanges which transforms the database underlying $y$ into the database underlying $x$. We consider the following setting:

- The mechanisms $M$ under investigation should provide $\varepsilon$-differential privacy and, given a node $x \in K$, they should return another node in $K$ (the choice of which depends on $M$’s internal randomization).

- The cost (= negated utility) of an output $y$, given input $x$, is defined as the distance between $x$ and $y$ in $G$, which is denoted as $\phi(x, y)$. We will refer to this distance measure as the record-exchange metric. Note that $|\phi(x, y) - \phi(x', y)| \leq 1$ holds for all $x, x', y \in K$ such that $x$ and $x'$ (resp. their underlying databases) are neighboring. Thus $-\phi$ (viewed as a utility function) has sensitivity 1.

Note that the record-exchange metric coincides with what is called “geodesic distance” with respect to the graph $G$ in some papers. We now present two examples where, in both cases, the record-exchange metric coincides with $1/2$ times the $L_1$-metric.

**Example 5.1** $(n, T)$-histograms. Suppose that the nodes in $G$ represent histograms with $n$ users and $T$ types of records (briefly called $(n, T)$-histograms hereafter), i.e., we may identify a node $x \in K$ with a vector $(\rho_1, \ldots, \rho_T)$ such that $n = \sum_{t=1}^T \rho_t$ and $\rho_t$ is the number of users whose record is of type $t$. Note that the record-exchange metric satisfies $\phi(x, y) = \frac{1}{2} \|x - y\|_1$ because
5.2. Optimal Mechanisms and Linear Programming

Each record-exchange can decrease the $L_1$-distance between two histograms by an amount of 2 (but not more).

**Example 5.2** (Sorted $(n,T)$-histograms). Suppose that the nodes in $G$ represent sorted $(n,T)$-histograms, i.e., we may identify a node $x \in K$ with a sorted sequence $\rho_1 \geq \ldots \geq \rho_T$ such that $\sum_{t=1}^T \rho_t = n$. Here $\rho_1$ (resp. $\rho_2$ and so on) denotes the number of users whose record occurs most often (resp. 2nd most often and so on) in the database.

Note that a node $x \in K$ representing a sorted $(n,n)$-histogram corresponds to the partition of the integer $n$ induced by a database $d$, which is the setting we consider in Chapter 4.

In this section, we investigate under which conditions the exponential mechanism is optimal in the sense of incurring the smallest possible expected error (measured in terms of the record-exchange metric) where expectation is taken over the (uniformly distributed) inputs $x \in K$ and over the internal randomization of the mechanism. We start by introducing several linear programs. The optimal solution of the first linear program we consider, denoted $\text{LP}^G[5.5]$ below, corresponds to the solution of the optimal $\varepsilon$-differentially private mechanism. Another linear program, denoted $\text{LP}^G[5.9]$ below, has an optimal solution which coincides with the one given by the exponential mechanism. We then provide some regularity conditions on the graph $G$ under which an optimal solution of $\text{LP}^G[5.9]$ also optimizes $\text{LP}^G[5.5]$, so that the exponential mechanism is optimal whenever the regularity conditions are valid.

We can now continue with our general discussion. Note that a mechanism $M$ with inputs and outputs taken from $K$ is formally given by probability parameters $p(y|x)$ denoting the probability of returning $y \in K$ when given $x \in K$ as input. Since, for each $x$, $p(y|x)$ is a distribution on $K$, we have

$$
(\forall x,y \in K : p(y|x) \geq 0) \land \left( \forall x \in K : \sum_{y \in K} p(y|x) = 1 \right).
$$

(5.1)

Moreover, if $M$ provides $\varepsilon$-differential privacy, we have

$$
\forall y \in K, \forall \{x,x'\} \in \mathcal{E} : p(y|x') \geq e^{-\varepsilon} \cdot p(y|x).
$$

(5.2)

Conversely, every choice of these probability parameters that satisfies (5.1) and (5.2) represents a mechanism that provides $\varepsilon$-differential privacy.

Suppose that $M$ is given by its probability parameters $p = (p(y|x))$ as described above. The average distance between $x \in K$ and the output $y \in K$, returned by $M$ when given $x$ as input, is then given as follows:

$$
\text{obj}^G(p) = \frac{1}{K} \cdot \sum_{x \in K} \sum_{y \in K} p(y|x) \phi(x, y).
$$

(5.3)
Chapter 5. Average-case Optimality of the Exponential Mechanism

Let $S_\eta = S_\eta(y)$ denote the set of all nodes in $K$ with distance $\eta$ to $y$ (the $\eta$-th layer of $G$ with respect to the start node $y$). Then

\[
\text{obj}_G^G(p) = \frac{1}{K} \cdot \sum_{y \in K} \text{obj}^G_y(p) \quad \text{for} \quad \text{obj}^G_y(p) = \sum_{\eta=0}^{D} \sum_{x \in S_\eta(y)} p(y|x) \eta.
\]

We pursue the goal to find an $\varepsilon$-differentially private mechanism $M$ that minimizes $\phi(x,y)$ on average. For this reason, we say that a mechanism $M^\star$ with probability parameters $p^\star$ is optimal with respect to $G$ if $p^\star$ is a minimizer of $\text{obj}_G^G(p)$ among all $p$ that satisfy (5.1) and (5.2). It is obvious from our discussion that the probability parameters $p^\star(y|x)$ representing an optimal mechanism with respect to $G$ are obtained by solving the following linear program:

\[
\begin{align*}
\text{minimize} & \quad \text{obj}_G^G(p) \\
\text{subject to} & \quad (5.1) \text{ and } (5.2) \\
& \quad \text{(LP}_G^{5.5})
\end{align*}
\]

We now bring into play the following modifications of condition (5.1):

\[
(\forall x, y \in K : p(y|x) \geq 0) \quad \land \quad \left( \sum_{x \in K} \sum_{y \in K} p(y|x) = K\right). \quad (5.6)
\]

\[
(\forall x, y \in K : p(y|x) \geq 0) \quad \land \quad \left( \forall y \in K : \sum_{x \in K} p(y|x) = 1\right). \quad (5.7)
\]

Note that (5.7) implies (5.6). Consider the following relatives of $\text{LP}_G^{5.5}$:

\[
\begin{align*}
\text{minimize} & \quad \text{obj}_G^G(p) \\
\text{subject to} & \quad (5.6) \text{ and } (5.2) \\
& \quad \text{(LP}_G^{5.8})
\end{align*}
\]

and

\[
\begin{align*}
\text{minimize} & \quad \text{obj}_G^G(p) \\
\text{subject to} & \quad (5.7) \text{ and } (5.2) \\
& \quad \text{(LP}_G^{5.9})
\end{align*}
\]

Given a graph $G = (K, E)$, a permutation $\sigma$ of $K$ is called automorphism if, for all $x, y \in K$, \{x, y\} $\in E$ $\Leftrightarrow$ \{\sigma(x), \sigma(y)\} $\in E$. The set of all automorphisms of $K$, under the operation of composition of functions, forms a group called the automorphism group of $G$. Such a group is called transitive if, for every $x, y \in K$, there exists an automorphism $\sigma$ of $K$ such that $\sigma(x) = y$.

\textbf{Lemma 5.1.} Suppose that the graph $G$ has a transitive automorphism group. Then every feasible solution $p$ for $\text{LP}_G^{5.8}$ can be transformed into another feasible solution $p'$ such that $\text{obj}^G(p') \leq \text{obj}^G(p)$ and $p'$ satisfies (5.7).
5.2. Optimal Mechanisms and Linear Programming

Proof. Let $\mathbf{p}$ be any feasible solution for LP$\mathbf{G}$\[5.8\]. For every $y \in \mathcal{K}$, let $K_y(\mathbf{p}) = \sum_{x \in \mathcal{K}} p(y|x)$. According to \[5.6\], we have $\sum_{y \in \mathcal{K}} K_y(\mathbf{p}) = K$. Define

$$\bar{p}(y|x) = \frac{1}{K_y(\mathbf{p})} p(y|x) \quad \text{and} \quad \overline{\text{obj}}_y(\mathbf{p}) = \sum_{\eta=0}^{D} \sum_{x \in S_{\eta}(y)} \bar{p}(y|x) \eta$$

and note that $\bar{p}$ satisfies \[5.2\] and \[5.7\]. We may now write $\text{obj}^G(\mathbf{p})$ as follows:

$$\text{obj}^G(\mathbf{p}) = \sum_{y \in \mathcal{K}} \frac{K_y(\mathbf{p})}{K} \cdot \overline{\text{obj}}_y(\mathbf{p})$$

Thus obj$^G(\mathbf{p})$ can be interpreted as the average of the cost terms $\underline{\text{obj}}_y(\mathbf{p})$ where the term $\overline{\text{obj}}_y(\mathbf{p})$ is chosen with probability $K_y(\mathbf{p})/K$. According to the pigeonhole principle, there exists $y^* \in \mathcal{K}$ such that $\overline{\text{obj}}_{y^*}(\mathbf{p}) \leq \text{obj}^G(\mathbf{p})$. Our strategy is to use the automorphism of $\mathcal{G}$ for building a new (and superior) feasible solution $\mathbf{p}'$ whose components contain $K$ duplicates of the parameter collection $(\bar{p}(y^*|x)_{x \in \mathcal{K}})$. To this end, let $\sigma_y$ be the automorphism which maps $y$ to $y^*$ and define

$$p'(y|x) = \bar{p}(y^*|\sigma_y(x))$$

Note that $x \in S_{\eta}(y)$ if and only if $\sigma_y(x) \in S_{\eta}(y^*)$. Obviously, $\mathbf{p}' \geq 0$ and, for every $y \in \mathcal{K}$, we have

$$K_y(\mathbf{p}') = \sum_{x \in \mathcal{K}} p'(y|x) = \sum_{x \in \mathcal{K}} \bar{p}(y^*|\sigma_y(x)) = \sum_{x \in \mathcal{K}} \bar{p}(y^*|x) = 1$$

This shows that $\mathbf{p}'$ satisfies \[5.7\]. Moreover, $\mathbf{p}'$ satisfies \[5.2\] since, for every $y \in \mathcal{K}$ and every $\{x, x'\} \in \mathcal{E}$, we have

$$e^{-\varepsilon} \cdot p'(y|x) = e^{-\varepsilon} \cdot \bar{p}(y^*|\sigma_y(x)) \leq \bar{p}(y^*|\sigma_y(x')) = p'(y|x')$$

where the inequality follows from the fact that $\bar{p}$ satisfies \[5.2\] and $\sigma_y$ is an automorphism. The following calculation shows that $\text{obj}_y(\mathbf{p}') = \overline{\text{obj}}_{y^*}(\mathbf{p})$ holds for every $y \in \mathcal{K}$:

$$\text{obj}_y(\mathbf{p}') = \sum_{\eta=0}^{D} \sum_{x \in S_{\eta}(y)} p'(y|x) \eta$$

$$= \sum_{\eta=0}^{D} \sum_{x \in S_{\eta}(y)} \bar{p}(y^*|\sigma_y(x)) \eta$$

$$= \sum_{\eta=0}^{D} \sum_{x \in S_{\eta}(y^*)} \bar{p}(y^*|x) \eta$$

$$= \overline{\text{obj}}_{y^*}(\mathbf{p})$$
We now obtain
\[
\operatorname{obj}^G(p') = \frac{1}{K} \cdot \sum_{y \in K} \operatorname{obj}_y(p') = \overline{\operatorname{obj}}_y(p) \leq \operatorname{obj}^G(p),
\]
which concludes the proof.

The following result is an immediate consequence of Lemma 5.1.

**Corollary 5.2.** The optimal values of the problems \(\text{LP}^G[5.8]\) and \(\text{LP}^G[5.9]\) coincide. Moreover, every optimal solution for \(\text{LP}^G[5.9]\) is an optimal solution for \(\text{LP}^G[5.8]\).

We say that the graph \(G\) has a regular layer sequence with respect to \(y \in K\) if, for all \(\eta\) and for all \(x,x' \in S_\eta(y)\), the nodes \(x\) and \(x'\) have the same number of neighbors in \(S_{\eta-1}(y)\) and the same number of neighbors in \(S_{\eta+1}(y)\). Let \(E[y] = E \cap \bigcup_{\eta=0}^{N-1} (S_\eta(y) \times S_{\eta+1}(y))\), i.e., \(E[y]\) contains the edges in \(E\) which connect two nodes in subsequent layers (but excludes the edges which connect two nodes in the same layer).

**Lemma 5.3.** Suppose that the graph \(G = (K,E)\) has a transitive automorphism group and a regular layer sequence w.r.t. any \(y \in K\). Then the problems \(\text{LP}^G[5.8]\) and \(\text{LP}^G[5.9]\) have an optimal solution that satisfies
\[
\forall y \in K, \forall (x,x') \in E[y] : p(y|x') \geq e^{-\varepsilon} \cdot p(y|x) \tag{5.10}
\]
with equality.

**Proof.** The problem \(\text{LP}^G[5.9]\) decomposes into \(K = |K|\) independent subproblems, one subproblem \(\text{LP}^G(y)\) for each fixed choice of \(y \in K\):

\[
\begin{align*}
\text{minimize} \quad & \sum_{y \in K} \operatorname{obj}_y(p) = \sum_{\eta=0}^{D} \left( \sum_{x \in S_\eta(y)} p(y|x) \right) \\
\text{subject to} \quad & p \geq 0, \\
& \sum_{x \in K} p(y|x) = 1 \quad \text{and} \\
& \forall \{x,x'\} \in E : p(y|x') \geq e^{-\varepsilon} \cdot p(y|x).
\end{align*}
\]

Let \(\text{LP}^*\) (the notation will become clear in Section 5.3) be the linear program that is obtained from \(\text{LP}^G(y)\) by substituting the weaker constraint
\[
\forall (x,x') \in E[y] : p(y|x') \geq e^{-\varepsilon} \cdot p(y|x)
\]
for
\[
\forall \{x,x'\} \in E : p(y|x') \geq e^{-\varepsilon} \cdot p(y|x).
\]

In Section 5.3 we will prove the following result:
5.2. Optimal Mechanisms and Linear Programming

Claim 5.4. If $G[y] = (K, E[y])$ has a regular layer sequence, then $LP^{**}$ has an optimal solution with the following properties:

1. The parameter vector $(p(y|x))_{x \in K}$ (with a fixed choice of $y$) assigns the same probability mass to all nodes $x$ taken from the same layer.

2. For every $(x, x') \in E[y]$, it satisfies the constraint $p(y|x') \geq e^{-\varepsilon} \cdot p(y|x)$ with equality.

It immediately follows that this optimal solution is also an optimal solution for $LP^G[y]$, which completes the proof.

The proof of Claim 5.4 is lengthy and will therefore be given later. See Lemma 5.7 in Section 5.3. Recall that $\phi(x, y)$ denotes the distance between $x$ and $y$ with respect to the record-exchange metric. Here comes the main result of this section which essentially states that the exponential mechanism is optimal under the assumptions made in Lemma 5.3.

Theorem 5.5. Under the same assumptions as in Lemma 5.3, the following holds. An optimal mechanism for $LP^G[5.5]$ (and even for $LP^G[5.8]$ and for $LP^G[5.9]$) is obtained by setting

$$\forall x, y \in K : p(y|x) \propto \exp(-\varepsilon \cdot \phi(x, y)) .$$

Proof. Let $p$ be the optimal solution for $LP^G[5.8]$ and $LP^G[5.9]$ that satisfies (5.10) with equality so that

$$\forall y \in K, \forall (x, x') \in E[y] : p(y|x') = e^{-\varepsilon} \cdot p(y|x) .$$

Unrolling this recursion, we get

$$p(y_0|x_0) = \frac{\exp(-\varepsilon \cdot \phi(x_0, y_0))}{\sum_{x \in K} \exp(-\varepsilon \cdot \phi(x, y_0))} .$$

The transitivity of the automorphism group of $G$ implies that

$$\forall x_0, y_0 \in K : \sum_{x \in K} \exp(-\varepsilon \cdot \phi(x, y_0)) = \sum_{y \in K} \exp(-\varepsilon \cdot \phi(y, x_0)) .$$

It follows that $p(y_0|x_0) = p(x_0|y_0)$. As a feasible solution of $LP^G[5.9]$, $p$ satisfies (5.7). Since $p(y_0|x_0) = p(x_0|y_0)$, it must also satisfy (5.1). Thus $p$ is a feasible solution for $LP^G[5.5]$. Since it is even optimal among the feasible solutions of the relaxation $LP^G[5.8]$, it must be optimal for $LP^G[5.5]$. \qed
5.3 Proof of Claim 5.4 and Additional Remarks on [LP**]

Recall that $G = (K, E)$ denotes a graph with $K = |K|$ nodes and diameter $D$. Fix some $y \in K$ and call it the “start node”. Recall that $S_\eta(y) = S_\eta(y)$ is the set of all nodes in $K$ with distance $\eta$ to $y$ (the $\eta$-th layer in $G$). The cardinality of $S_\eta(y)$ is denoted by $s_\eta(y)$, or simply by $s_\eta$. For instance, $S_0 = \{y\}$ and $S_1$ is the set of all neighbors of $y$ in $G$. An edge $e \in E$ either connects two nodes in subsequent layers or it connects two nodes in the same layer. Let again $E[y] \subseteq E$ be the set of edges of the former kind and let $G[y] = (K, E[y])$. In other words, $G[y]$ is the layered graph that contains all shortest paths to the start node $y$. We consider an edge in $E[y]$ as being directed away from $y$, i.e., $(x, x') \in E[y]$ implies that $x \in S_\eta$ and $x' \in S_{\eta+1}$ for some $\eta \in \{0, 1, \ldots, D - 1\}$. Note that $E[y]$ naturally partitions into the (disjoint) union of $E_0, E_1, \ldots, E_{D-1}$ where $E_\eta = E[y] \cap (S_\eta \times S_{\eta+1})$. Let $0 < \kappa < 1$ denote a constant scaling factor.

In this section, we consider the following two linear optimization problems:

\[
\begin{align*}
\text{minimize} \quad & \quad p \in (p_\eta) \quad \text{obj}_1(p) = \sum_{\eta=0}^D s_\eta p_\eta \eta \\
\text{subject to} \quad & \quad p \geq 0, \\
& \quad \sum_{\eta=0}^D s_\eta p_\eta = 1 \quad \text{and} \\
& \quad \forall \eta \in \{0, \ldots, D - 1\} : p_{\eta+1} \geq \kappa \cdot p_\eta ;
\end{align*}
\]

and

\[
\begin{align*}
\text{minimize} \quad & \quad p \in (p_x) \quad \text{obj}_2(p) = \sum_{\eta=0}^D \left( \sum_{x \in S_\eta} p_x \right) \eta \\
\text{subject to} \quad & \quad p \geq 0, \\
& \quad \sum_{x \in K} p_x = 1 \quad \text{and} \\
& \quad \forall (x, x') \in E[y] : p_{x'} \geq \kappa \cdot p_x .
\end{align*}
\]

In other words, we would like to find a probability distribution on $K$ that minimizes the average distance to the start node $y$ subject to the constraints in $\text{LP}^*$ (resp. $\text{LP}^{**}$). In $\text{LP}^{**}$ we can assign individual probabilities to all nodes whereas, in $\text{LP}^*$ we have to assign the same probability $p_\eta$ to all nodes in the $\eta$-th layer $S_\eta$ (so that the total probability mass assigned to $S_\eta$ equals $s_\eta p_\eta$). Note that $\text{LP}^{**}$ yields the problem that occurs under the same name in the proof of Lemma 5.3 provided that we set $\kappa = e^{-\varepsilon}$ and $p_x = p(y|x)$.

As for $\text{LP}^*$ it is intuitively clear that we should move as much probability mass as possible to layers close to the start node $y$.

**Lemma 5.6.** $\text{LP}^*$ is bounded and feasible. Moreover, there is a unique optimal solution that satisfies all constraints in $\text{LP}^*$ with equality.
5.3. Proof of Claim 5.4 and Additional Remarks on \( \text{LP}^\star \)

**Proof.** \( \text{LP}^\star \) is clearly bounded. A feasible solution can be obtained as follows. Pick any \( v > 0 \) and, for \( \eta = 0, \ldots, D \), tentatively set \( p_\eta = \kappa^\eta \cdot v \). Then \( p \geq 0 \) and the constraint \( \sum_{\eta=0}^{D} p_\eta = 1 \) can now be satisfied by virtue of normalization.\(^4\)

The proof can now be accomplished by showing that any feasible solution, say \( p = (p_\eta) \), that does not satisfy every constraint in \( \text{LP}^\star \) with equality can be improved. To this end, pick any \( p \) such that \( p_{l+1} = \kappa p_l + a \) for some \( a > 0 \). The idea for getting a new feasible solution, say \( p' = (p'_\eta) \), that is superior to \( p \) is quite simple. We move a (carefully chosen) fraction of the total probability mass \( s_{l+1} p_{l+1} \), currently assigned to \( S_{l+1} \), from \( S_{l+1} \) to \( S_l \) where it will be evenly shared by the \( s_l \) nodes in \( S_l \). The only possible \( \text{LP}^\star \) constraint that could be possibly violated by this maneuver is actually the constraint that involves layers \( l \) and \( l+1 \). However, since this constraint currently has some slack \( a \), we will not run into troubles as long as a sufficiently small amount of probability mass is transferred. A possible implementation of this plan is as follows. Set

\[
\left( a' = \frac{s_l}{\kappa s_{l+1} + s_l} \cdot a \right) \land \left( p'_l = p_l + \frac{s_{l+1}}{s_l} \cdot a' \right) \land \left( p'_{l+1} = p_{l+1} - a' = \kappa p_l + a - a' \right)
\]

and \( p'_\eta = p_\eta \) for every \( \eta \in \{0, \ldots, D-1\} \setminus \{l, l+1\} \). Clearly \( p' \geq 0 \). Note that

\[
p'_{l+1} = \kappa p_l + a - a' = \kappa p'_l - \frac{\kappa s_{l+1}}{s_l} a' + a - a' = \kappa p'_l + a - \left(1 + \frac{\kappa s_{l+1}}{s_l}\right) a' = \kappa p'_l + a - \frac{\kappa s_{l+1} + s_l}{s_l} a' = \kappa p'_l .
\]

Thus \( p' \) satisfies all constraints in \( \text{LP}^\star \). Note furthermore that

\[
s_{l+1} p'_{l+1} + s_l p'_l = (s_{l+1} p_{l+1} - s_{l+1} a') + (s_l p_l + s_{l+1} a') = s_{l+1} p_{l+1} + s_l p_l .
\]

It follows that \( \sum_{\eta=0}^{D} s_\eta p'_\eta = \sum_{\eta=0}^{D} s_\eta p_\eta = 1 \). We may now conclude that \( p' \) is a feasible solution for \( \text{LP}^\star \). The next calculation shows that \( p' \) has smaller cost than \( p \):

\[
(l+1) \cdot s_{l+1} p'_{l+1} + l \cdot s_l p'_l = (l+1) \cdot (s_{l+1} p_{l+1} - s_{l+1} a') + l \cdot (s_l p_l + s_{l+1} a') = (l+1) \cdot s_{l+1} p_{l+1} + l \cdot s_l p_l - s_{l+1} a' .
\]

These observations complete the proof. \( \square \)

\(^4\)Note that the constraints in \( \text{LP}^\star \) cannot become violated when every parameter \( p_\eta \) is scaled by a constant factor.
Recall that \( G \) with start node \( y \) is said to have a regular layer sequence if nodes in the same layer of \( G[y] \) have the same in-degree and the same out-degree. The next result is essentially a reformulation of Claim 5.4 from Section 5.2.

**Lemma 5.7.** \( \text{LP}^{\star\star} \) is bounded and feasible. Moreover, if \( G[y] = (\mathcal{K}, \mathcal{E}[y]) \) has a regular layer sequence, then \( \text{LP}^{\star\star} \) has an optimal solution that, first, assigns the same probability mass to all nodes in the same layer, and, second, satisfies all constraints with equality.

**Proof.** Clearly \( \text{LP}^{\star\star} \) is bounded. The existence of a feasible solution can be shown in the same (easy) way as it is done in the proof of Lemma 5.6. Thus we only have to show that \( \text{LP}^{\star\star} \) has an optimal solution that satisfies all constraints with equality. Call a feasible solution \( p = (p_x) \) of \( \text{LP}^{\star\star} \) normalized if \( p \) assigns the same probability mass to all nodes in the same layer, say \( p_x = \bar{p}_\eta \) for every node \( x \) in layer \( \eta \). As for normalized feasible solutions, \( \text{LP}^{\star\star} \) collapses to \( \text{LP}^\star \). According to Lemma 5.6, there is a unique optimal solution among all normalized feasible solutions of \( \text{LP}^{\star\star} \) that satisfies all constraints in \( \text{LP}^{\star\star} \) with equality.\(^5\) Thus, we now have to show that every feasible solution can be normalized without increasing its cost. To this end, let \( p = (p_x) \) denote a fixed but arbitrary feasible solution for \( \text{LP}^{\star\star} \). For \( \eta = 0, \ldots, D \), we set \( \bar{p}_\eta = \frac{1}{s_\eta} \sum_{x \in S_\eta} p_x \), i.e., \( \bar{p}_\eta \) is the probability mass assigned by \( p \) to nodes in \( S_\eta \) on average. We claim that setting \( p'_x = \bar{p}_\eta \) for every node \( x \in S_\eta \) yields a normalized feasible solution of the same cost as \( p \). Clearly \( p' \geq 0 \). Moreover \( \sum_{x \in \mathcal{K}} p'_x = \sum_{x \in \mathcal{K}} p_x = 1 \) because \( p \mapsto p' \) leaves the total probability mass assigned to any layer \( S_\eta \) unchanged. For the same reason the cost of \( p' \) coincides with the cost of \( p \), i.e., \( \text{obj}_2(p') = \text{obj}_2(p) \). It remains to show that \( p' \) satisfies the constraints in \( \text{LP}^{\star\star} \). To this end, pick any \( \eta \in \{0, \ldots, D-1\} \) and any \((x, x') \in \mathcal{E}_\eta \). Let \( t_\eta \to \eta \) denote the out-degree of \( x \) (or of any other node from \( S_\eta \)) and let \( t_{\eta+1} \leftarrow \eta \) denote the in-degree of \( x' \) (or of any other node from \( S_{\eta+1} \)). A simple double counting argument shows that

\[
|\mathcal{E}_\eta| = s_\eta t_\eta \to \eta = s_{\eta+1} t_{\eta+1} \leftarrow \eta . \tag{5.11}
\]

The following calculation shows that \( p'_{x'} \geq \kappa p'_{x} \):

\[
p'_{x'} &= \frac{1}{s_{\eta+1}} \cdot \sum_{v \in S_{\eta+1}} p_v \\
&\geq \frac{1}{s_{\eta+1} t_{\eta+1} \leftarrow \eta} \cdot \sum_{v \in S_{\eta+1}} \sum_{u \in S_\eta, (u,v) \in \mathcal{E}_\eta} p_v \\
&\geq \frac{1}{s_\eta t_\eta \to \eta} \cdot \sum_{u \in S_\eta} \sum_{v \in S_{\eta+1} \cup \mathcal{E}_\eta} p_v \tag{5.11}
\]

\(^5\)The constraints in \( \text{LP}^{\star\star} \) coincide with those in \( \text{LP}^\star \) for normalized feasible solutions.
5.3. Proof of Claim 5.4 and Additional Remarks on LP*

\[ \geq \kappa \cdot \frac{1}{s_{\eta+1}} \cdot \sum_{u \in S_{\eta}} \sum_{v : (u,v) \in E_{\eta}} p_u \]

\[ = \kappa \cdot \frac{1}{s_{\eta}} \cdot \sum_{u \in S_{\eta}} p_u \]

The equations marked "\(*\)"] make use of the assumption that \( G[\eta] \) has a regular layer sequence. The whole discussion can be summarized by saying that \( p' \) is a normalized feasible solution for LP** and its cost equals the cost of the feasible solution \( p \) that we started with. This concludes the proof. \( \square \)

Let LP* and LP** denote the optimization problems that result from LP* and LP** respectively, when the underlying graph \( G = (K, E) \) has infinitely many nodes so that the layered graph \( G[\eta] = (K, E[\eta]) \) might have infinitely many layers \( S_0, S_1, S_2, \ldots \). In the formal definition of LP* and LP**, we only have to substitute \( \infty \) for \( D \). An inspection of the proofs of Lemmas 5.6 and 5.7 reveals that they hold, mutatis mutandis, for the problems LP* and LP** as well.

**Corollary 5.8.** LP* and LP** are bounded and feasible. Moreover, there is a unique optimal solution for LP* that satisfies all of its constraints with equality and, if \( G[\eta] = (K, E[\eta]) \) has a regular layer sequence, then LP** has an optimal solution that satisfies all of its constraints with equality.

**Example 5.3.** Let \( G_1 \) be an infinite path \( y_0, y_1, y_2, \ldots \) with start node \( y_0 \). It follows from Corollary 5.8 that LP** has an optimal solution that satisfies all constraints with equality. This leads to the following average distance from \( y_0 \):

\[
\frac{\sum_{\eta \geq 1} \kappa \eta \eta}{\sum_{\eta \geq 0} \kappa \eta} = \frac{\kappa}{1 - \kappa^2} = \frac{\kappa}{1 - \kappa}.
\]

Let \( G_2 \) be the graph consisting of two infinite paths, \( y_0, y_{-1}, y_{-2}, \ldots \) and \( y_0, y_1, y_2, \ldots \) both of which are starting from the start node \( y_0 \). Again Corollary 5.8 applies and the optimal average distance from \( y_0 \) is calculated as follows:

\[
2 \cdot \frac{\sum_{\eta \geq 1} \kappa \eta \eta}{1 + 2 \cdot \sum_{\eta \geq 1} \kappa \eta} = \frac{2\kappa}{1 + 2\kappa} = \frac{2\kappa}{1 - \kappa^2}.
\]

As for finite paths, we have the following result:

**Lemma 5.9.** Let \( P_l \) be a path of length \( 2l \) and let \( y_0 \) be the start node located in the middle of \( P_l \). Let \( \varphi(l) \) denote the optimal value that can be achieved in the linear program LP** w.r.t. to \( G = P_l \). Then the following holds:

\[
\frac{\varphi(l)}{\sum_{x \in S(l)} p_x} = \frac{2\kappa}{1 + 2\kappa}.
\]

\( \square \)
Chapter 5. Average-case Optimality of the Exponential Mechanism

(i) \( \mathbf{P}^{\ast\ast} \) has an optimal solution that satisfies all constraints with equality so that

\[
\varphi(l) = \frac{2 \cdot \sum_{\eta=1}^{l} \kappa^\eta \eta}{1 + 2 \cdot \sum_{\eta=1}^{l} \kappa^\eta}.
\]

(ii) The function \( \varphi(l) \) is strictly increasing with \( l \).

(iii) We have

\[
\varphi(l) > \frac{2\kappa}{1 - \kappa^2} \cdot \left(1 - \kappa^l - l\kappa^l(1 - \kappa)\right).
\]

Moreover, if \( l \geq s/(1 - \kappa) \), then

\[
\varphi(l) > \frac{2\kappa}{1 - \kappa^2} \cdot \left(1 - (s + 1)e^{-s}\right).
\]

(iv) \( \lim_{l \to \infty} \varphi(l) = \frac{2\kappa}{1 - \kappa^2} \).

Proof. Let \( P_l = y_{-l}, \ldots, y_{-1}, y_0, y_1, \ldots, y_l \).

(i) Lemma 5.7 applies because \( P_l[y_0] \) has a regular layer sequence.

(ii) An optimal solution for \( P_{l+1} \) can be transformed into a feasible solution for \( P_l \) by transferring the probability mass of the nodes \( y_{-(l+1)}, y_{l+1} \) to the nodes \( y_{-l}, y_l \), respectively. This transfer strictly reduces the cost. The optimal cost \( \varphi(l) \) that can be achieved on \( P_l \) is, in turn, smaller than the cost of this feasible solution.

(iii) We start with the following calculation:

\[
\sum_{\eta=1}^{l} \kappa^{\eta-1} \eta = \sum_{\eta=1}^{l} \kappa^{\eta-1} \eta - \sum_{\eta=1}^{l+1} \kappa^{\eta-1} \eta
\]

\[
= \frac{1}{(1 - \kappa)^2} - \kappa^l \cdot \sum_{\eta=1}^{l} \kappa^{\eta-1} (\eta + l)
\]

\[
= \frac{1}{(1 - \kappa)^2} - \kappa^l \cdot \left(\frac{1}{(1 - \kappa)^2} + \frac{l}{1 - \kappa}\right)
\]

\[
= \frac{1}{(1 - \kappa)^2} \cdot \left(1 - \kappa^l - l\kappa^l(1 - \kappa)\right)
\]

Setting \( F = 1 - \kappa^l - l\kappa^l(1 - \kappa) \), it follows that

\[
\varphi(l) = \frac{2\kappa}{1 - \kappa^2} \cdot \frac{F}{1 + 2 \cdot \sum_{\eta=1}^{l} \kappa^\eta} > \frac{2\kappa}{1 - \kappa^2} \cdot F.
\]

Since the latter expression differs from (5.12) by the factor \( F \) only, we obtain (5.14). The function \( s \mapsto (s + 1)e^{-s} \) is strictly monotonically
decreasing for all \( s \geq 0 \). It therefore suffices to verify the bound (5.15) for \( s = (1 - \kappa)l \) so that

\[
\kappa^l + l\kappa^l(1 - \kappa) = \kappa^l(s + 1).
\]

Noting that

\[
\kappa^l = \kappa^{s/(1-\kappa)} = (1 - (1 - \kappa))^{s/(1-\kappa)} < e^{-s},
\]

we may conclude that \( \kappa^l + l\kappa^l(1 - \kappa) < (s + 1)e^{-s} \). From this, in combination with (5.14), the bound (5.15) is immediate.

(iv) The fourth assertion of Lemma 5.9 is immediate from the third one.

\(\Box\)

Even though the regularity conditions for \( G \) (transitive automorphism group and regular layer sequence) are satisfied in simple settings (for instance, when each node in the graph corresponds to a binary database of size \( n \)), we do not expect this to be the case in most applications. For example, the regularity conditions are not fully satisfied by the graph representing sorted \((n, T)\)-histograms introduced in Example 5.2. However, we conjecture, first, that these conditions are approximately satisfied for very large databases and, second, that the exponential mechanism is still approximately optimal when these conditions hold approximately. At the time being, we are not able to formally verify this conjecture for graphs \( G \) of practical interest. In Section 5.4 we illustrate the kind of arguments that could be brought into play by presenting a very precise analysis for the simple case where the graph \( G \) actually is a long but finite path. Developing these arguments further so as to analyze more reasonable classes of graphs, e.g., graphs representing the neighborhood relation for sorted histograms, remains an open problem.

5.4 A Toy Example: the Path Graph

Throughout this section, we consider the graph \( G = (\mathcal{K}, \mathcal{E}) \) whose nodes \( y_1, \ldots, y_K \) form a path of length \( K - 1 \). Note that \( G \) does not satisfy the regularity condition: neither has \( G \) a transitive automorphism group nor has \( G[y] \) a regular layer sequence (except for \( y \) being chosen as one of the endpoints and, if \( K \) is odd, for \( y \) being chosen as the point in the middle of the path). Let \( \text{OPT}^G \) denote the smallest cost of a feasible solution for \( LP^G[5.5] \). We will show in this section that, despite the violation of the regularity condition, the exponential mechanism comes close to optimality provided that \( K \) is “sufficiently large”. The main idea for proving this is as follows. We will split the set of nodes into
Chapter 5. Average-case Optimality of the Exponential Mechanism

a “central part” (nodes separated away from the endpoints of the path) and a “peripheral part” (nodes located close to the endpoints). We then make use of the fact that all $\varepsilon$-differentially private mechanisms are on the horns of the following dilemma:

- If a feasible solution $p = (p(y|x))_{x,y \in K}$ puts much probability mass on peripheral nodes $y$, then the cost contribution of the terms $p(y|x)$ with $y$ “peripheral” and $x$ “central” will be large.

- If not, then the cost contribution of the terms $p(y|x)$ with $y$ “central” will be large. The proof of this statement will exploit the fact that, if $y$ has distance at least $l$ to both endpoints of the path, then $G[y]$ contains the path $P_l$ from Lemma 5.9 (with $y$ located in the middle of $P_l$) as a subgraph. It is then easy to argue that the term $\varphi(l)$ from Lemma 5.9 serves as a lower bound on the cost achieved by $p$.

We will now formalize these ideas. Let $l \geq 1$ be arbitrary but fixed. We assume that $K \geq 4l$. We define the following sets of “peripheral” nodes:

$$
\mathcal{K}_1 = \{y_1, \ldots, y_l\} \cup \{y_{K-l+1}, \ldots, y_K\} \quad \text{and} \quad \mathcal{K}_2 = \{y_1, \ldots, y_{2l}\} \cup \{y_{K-2l+1}, \ldots, y_K\} .
$$

In other words, $\mathcal{K}_1$ (resp. $\mathcal{K}_2$) contains all nodes that have a distance of at most $l - 1$ (resp. $2l - 1$) to one of the endpoints $y_1$ and $y_K$. The complements of these sets are denoted $\bar{\mathcal{K}}_1$ and $\bar{\mathcal{K}}_2$, respectively. Note that each node in $\bar{\mathcal{K}}_1$ (resp. $\bar{\mathcal{K}}_2$) has a distance of at least $l$ (resp. $2l$) to both endpoints. Moreover, any point in $\mathcal{K}_1$ has distance of at least $l$ to any node in $\bar{\mathcal{K}}_2$. For every set $W \subseteq \mathcal{K} \times \mathcal{K}$, we define

$$
\pi(W) = \sum_{(x,y) \in W} p(x,y) = \frac{1}{K} \cdot \sum_{(x,y) \in W} p(y|x) ,
$$

i.e., $\pi(W)$ is the total probability mass assigned to pairs $(x,y) \in W$ if $x \in \mathcal{K}$ is uniformly distributed and $y$ has probability $p(y|x)$ conditioned to $x$. Then $\pi(\mathcal{K} \times \mathcal{K}_1)$ denotes the total probability assigned to pairs $(x,y)$ with $y \in \mathcal{K}_1$. The total mass of pairs from $\bar{\mathcal{K}}_2 \times \mathcal{K}_1$ can then be bounded from below as follows:

$$
\pi(\bar{\mathcal{K}}_2 \times \mathcal{K}_1) = \pi(\bar{\mathcal{K}}_2 \times \mathcal{K}) - \pi(\bar{\mathcal{K}}_2 \times \bar{\mathcal{K}}_1) \geq \pi(\bar{\mathcal{K}}_2 \times \mathcal{K}) - \pi(\mathcal{K} \times \bar{\mathcal{K}}_1)
= \left(1 - \frac{4l}{K}\right) - (1 - \pi(\mathcal{K} \times \mathcal{K}_1)) = \pi(\mathcal{K} \times \mathcal{K}_1) - \frac{4l}{K} .
$$
5.4. A Toy Example: the Path Graph

Since \( p(x, y) = p(y|x)/K \), we may rewrite the cost function \( \text{obj}^G(p) \) from (5.3) as follows:

\[
\text{obj}^G(p) = \sum_{(x, y) \in K \times K} p(x, y) \phi(x, y) .
\]

Since, as mentioned above already, \( \phi(x, y) \geq l \) holds for all pairs \((x, y) \in \overline{K}_2 \times K_1\), we obtain a first lower bound on \( \text{obj}^G(p) \):

\[
\text{obj}^G(p) \geq \pi(\overline{K}_2 \times K_1) \cdot l \geq \left( \pi(K \times K_1) - \frac{4l}{K} \right) \cdot l .
\] (5.16)

The lower bound (5.16) is induced by the elements \( y \) taken from the “peripheral region” \( K_1 \). In the next step, we derive a lower bound that is induced by the elements \( y \) taken from the “central region” \( \overline{K}_1 \). We remind the reader of the short notation

\[
K_y(p) = \sum_{x \in K} p(y|x) \quad \text{and} \quad \overline{\text{obj}}_y(p) = \sum_{x \in K} \frac{p(y|x)}{K_y(p)} \cdot \phi(x, y)
\]

and mention just another way of expressing the cost function:

\[
\text{obj}^G(p) = \sum_{y \in K} \frac{K_y(p)}{K} \cdot \overline{\text{obj}}_y(p) .
\] (5.17)

We set \( \bar{p}_y(x) = p(y|x)/K_y(p) \) and observe that \( \sum_{x \in K} \bar{p}_y(x) = 1 \). Let \( \varphi(l) \) be the function given by (5.13).

**Claim 5.10.** If \( y \in \overline{K}_1 \), then \( \overline{\text{obj}}_y(p) \geq \varphi(l) \).

**Proof.** Suppose that \( y \in \overline{K}_1 \). Then the subgraph \( G'[y] \) of \( G[y] \) that consists of all nodes with distance at most \( l \) from \( y \) forms a path of length \( 2l \) with \( y \) being located in the middle (like the graph \( P_l \) from Lemma 5.9). Since \( p \) is a feasible solution of \( \text{LP}^\ast \) from Section 5.3, it follows that \( (\bar{p}_y(x))_{x \in K} \) is a feasible solution of \( \text{LP}^\ast(I) \). Thus, \( \overline{\text{obj}}_y(p) \) is lower-bounded by the optimal value that can be achieved in \( \text{LP}^\ast(I) \) for the graph \( G'[y] \). We can transform \( \bar{p}_y \) into a feasible solution \( \bar{p}'_y \) for \( \text{LP}^\ast(I) \) with respect to \( G'[y] \) without increasing the cost by shifting the probability mass of the nodes with distance more than \( l \) from \( y \) to one or both of the nodes with distance exactly \( l \) from \( y \). Since \( G'[y] \simeq P_l \), we may invoke Lemma 5.9 and conclude that the cost of \( \bar{p}_y \) is lower-bounded by \( \varphi(l) \). Putting everything together, we finally get \( \overline{\text{obj}}_y(p) \geq \varphi(l) \).

In view of (5.17) and in view of the obvious identity

\[
\sum_{y \in \overline{K}_1} \frac{K_y(p)}{K} = \pi(K \times \overline{K}_1) = 1 - \pi(K \times K_1) ,
\]
the above claim, in combination with Lemma 5.9, immediately implies the following second lower bound on the cost function:

\[
\text{obj}^\mathcal{G}(p) \geq (1 - \pi(K \times K)) \cdot \frac{2\kappa}{1 - \kappa^2} \cdot \left(1 - \kappa^l - l\kappa^l(1 - \kappa)\right)
\]
\[
\geq (1 - \pi(K \times K)) \cdot \frac{2\kappa}{1 - \kappa^2} \cdot \left(1 - (s + 1)e^{-s}\right),
\]

(5.18)

where the final inequality is valid provided that \(l \geq s/(1 - \kappa)\). If \(\pi(K \times K) \geq 1/s\), we may invoke (5.16) and conclude that

\[
\text{obj}^\mathcal{G}(p) \geq \frac{1}{s - 4l} \cdot \frac{s}{1 - \kappa} = \left(1 - \frac{4sl}{K}\right) \cdot \frac{1}{1 - \kappa}.
\]

Otherwise, if \(\pi(K \times K) < 1/s\), we may invoke (5.18) and conclude that

\[
\text{obj}^\mathcal{G}(p) > \frac{2\kappa}{1 - \kappa^2} \cdot \left(1 - \frac{1}{s}\right) \cdot \left(1 - (s + 1)e^{-s}\right).
\]

We can summarize this discussion as follows.

**Theorem 5.11.** Let \(\mathcal{G} = (\mathcal{K}, \mathcal{E})\) be a path of length \(K - 1\). Suppose that \(s \geq 1\), \(0 < \kappa < 1\), \(l \geq s/(1 - \kappa)\) and \(K \geq 4l\). Then,

\[
\text{OPT}^\mathcal{G} \geq \frac{1}{1 - \kappa} \cdot \min \left\{1 - \frac{4sl}{K}, \frac{2\kappa}{1 + \kappa} \cdot \left(1 - \frac{1}{s}\right) \cdot \left(1 - (s + 1)e^{-s}\right)\right\}.
\]

**Corollary 5.12.** With the same notations and assumptions as in Theorem 5.11, the following holds. If \(s \geq 2\) and \(K \geq s^2l(1 + \kappa)/\kappa\), then

\[
\text{OPT}^\mathcal{G} \geq \frac{2\kappa}{1 - \kappa^2} \cdot \left(1 - \frac{1}{s}\right) \cdot \left(1 - (s + 1)e^{-s}\right).
\]

**Proof.** For \(s \geq 2\) and \(K \geq s^2l(1 + \kappa)/\kappa\) the minimum in Theorem 5.11 is taken by the second of the two possible terms. \(\square\)

In the sequel, we set \(\kappa = e^{-\varepsilon}\). We would like to show that the parameter vector \((p(y|x))\) which represents the exponential mechanism comes close to optimality. To this end, we need an upper bound on \(\text{obj}^\mathcal{G}(p)\). In a first step, we determine an upper bound on the cost induced by the exponential mechanism which makes \(p(y|x)\) proportional to \(\kappa^{\phi(x,y)} = \exp(-\varepsilon\phi(x,y))\). As shown in Section 3.3.2, this mechanism might achieve \(2\varepsilon\)-differential privacy only. In a second step, we determine an upper bound on the cost induced by the \(\varepsilon\)-differentially private exponential mechanism which makes \(p(y|x)\) proportional to \(\kappa^{\phi(x,y)/2} = \exp(-\varepsilon\phi(x,y)/2)\). But let us start with the first step.
5.4. A Toy Example: the Path Graph

Lemma 5.13. Suppose that the graph $G = (K, E)$ forms a path of length $K - 1$. If $p$ is determined by the $2\varepsilon$-differentially private exponential mechanism which makes $p(y|x)$ proportional to $\kappa^{\phi(x,y)}$, then

$$\text{obj}^G(p) < \frac{2\kappa}{1 - \kappa^2}.$$  

Proof. Consider the cost function $\text{obj}^G(p)$ in (5.3). It suffices to show that, for every $x \in K$, we have

$$\sum_{y \in K} p(y|x) \phi(x, y) < \frac{2\kappa}{1 - \kappa^2}.$$  

(5.19)

According to the definition of $p$ in Lemma 5.13, we have

$$p(y|x) = \frac{\kappa^{\phi(x,y)}}{\sum_{y' \in K} \kappa^{\phi(x,y')}}.$$  

Let $x$ be the $t$-th node on the path. Then

$$\sum_{y \in K} p(y|x) \phi(x, y) = \frac{\sum_{\eta=1}^{t-1} \kappa^\eta \eta + \sum_{\eta=1}^{K-t} \kappa^\eta \eta}{1 + \sum_{\eta=1}^{t-1} \kappa^\eta + \sum_{\eta=1}^{K-t} \kappa^\eta}.$$  

A straightforward computation shows that the latter expression coincides with $C\kappa/(1 - \kappa)$ for

$$C = \frac{2 - tk^t - (t - 1)k^t - (K - t + 1)k^{K-t} + (K - t)k^{K-t+1}}{1 + \kappa - k^t - k^{K-t+1}} < \frac{2}{1 + \kappa}.$$  

Combining these pieces of puzzle, we arrive at (5.19), as desired. \hfill \square

Note that the bound provided in Lemma 5.13 is optimal asymptotically, i.e., when $K$ and the slack parameters $l, s$ in Corollary 5.12 approach infinity. An application of Corollary 5.12 and of Lemma 5.13 (with $\kappa^{1/2} = \sqrt{\kappa} = e^{-\varepsilon/2}$ at the place of $\kappa = e^{-\varepsilon}$) immediately leads to the following result.

Corollary 5.14. Suppose that the graph $G = (K, E)$ forms a path of length $K - 1$, $s \geq 2$ and $K \geq s^2l(1 + \kappa)/\kappa$. If $p$ is determined by the $\varepsilon$-differentially private exponential mechanism which makes $p(y|x)$ proportional to $\kappa^{\phi(x,y)/2} = \exp(-\varepsilon\phi(x,y)/2)$, then

$$\frac{\text{OPT}^G}{\text{obj}^G(p)} \geq \frac{\kappa(1 - \sqrt{\kappa^2})}{\sqrt{\kappa(1 - \kappa^2)}} \cdot \left(1 - \frac{1}{s}\right) \cdot \left(1 - (s + 1)e^{-s}\right)$$

$$\geq \frac{\sqrt{\kappa}}{1 + \kappa} \cdot \left(1 - \frac{1}{s}\right) \cdot \left(1 - (s + 1)e^{-s}\right).$$
We conclude this section by noting that \( \sqrt{\kappa}/(1 + \kappa) \) is close to 1/2 if \( \kappa \) is close to 1.

## 5.5 Related Work

A general upper bound on the error introduced by the exponential mechanism can be found in the work of McSherry and Talwar [66]. Lower bounds in differential privacy have been extensively studied and a range of techniques for proving lower bounds have been introduced [43, 65, 23, 111, 26, 5]. The optimality of differentially private mechanisms has been the subject of recent studies. Kairouz et al. [52] introduce a family of mechanisms which contains a utility-maximizer under the local model of privacy. Koufogiannis et al. [60] investigate the optimality of the Laplace mechanism under the Lipschitz privacy framework. In particular, they show that the Laplace mechanism is optimal for identity queries in terms of the mean-squared error, when privacy is guaranteed with respect to the \( L_1 \)-norm. Geng et al. [38] show that the mean-squared error introduced by the staircase mechanism is optimal for low-dimensional queries.

Linear programming theory can be leveraged to show lower bounds on the error needed for achieving any meaningful privacy guarantee [31, 23]. Hsu et al. [46] investigate how to solve a linear program under differential privacy. Hardt and Talwar [43] exploit linear programming theory to show tight upper and lower bounds on the amount of noise needed to provide differential privacy for \( r \) linear queries on databases in \( \mathbb{R}^n \).

Our contribution is mostly related to the work of Ghosh et al. [39] and Brenner and Nissim [17]. In their paper, Ghosh et al. [39] consider Bayesian information consumers that wish to compute the number of entries in a database satisfying a given predicate. A Bayesian information consumer is characterized by a prior belief and a loss-function, which quantify the consumer’s side knowledge and the quality of the answer provided. Introducing a linear program modeling the privacy constraints, they show that a discrete variant of the Laplace mechanism enables optimality (after a deterministic post-processing of the output) for all Bayesian information consumers. Such a mechanism is usually referred to as universally optimal. In a follow up work, Brenner and Nissim [17] show that universally optimal mechanisms for Bayesian consumers are extremely rare, proving that they essentially exist only for a single counting query. Their proof makes use of a so-called privacy constraint graph, where the vertices correspond to the values of the output space, and the edges correspond to pairs of values resulting by applying the query function to neighboring databases. In contrast to [39] and [17], in this chapter we restrict our attention to a single

\[\text{(6)}\] This corresponds to the problem of computing a counting query (up to normalization) which is closely related to the content of Chapter 7.
information consumer who has a uniform prior over the input/output space and measures the loss in terms of the record-exchange metric. We then study under which conditions on the structure of the privacy constraint graph the solution of the optimal differentially private mechanism (modeled as a linear program similar to the one introduced by Ghosh et al. [39]) coincides with the solution that the exponential mechanism delivers.

5.6 Conclusions and Open Problems

In this chapter, we investigated under which conditions the average-case error—measured in terms of the record-exchange metric—introduced by the exponential mechanism is optimal. In particular, we showed that, if the input/output universe of the mechanism can be modeled as a graph which has a transitive automorphism group and a regular layer sequence, then the solution delivered by the exponential mechanism achieves the minimum average error. Additionally, we provided a toy example in which the optimality is preserved (up to a constant factor) even if these regularity conditions hold only to a certain extent. Even though the kind of arguments we employed in this simple example could prove useful towards a more general goal, demonstrating such a result for more interesting applications, e.g., graphs corresponding to sorted \((n, T)\)-histograms discussed in Example 5.2, remains a major open problem. We highlight that this problem is very much related to the problem of privately releasing integer partitions which we addressed in Chapter 4. In particular, showing the optimality (up to a constant factor) of the exponential mechanism for sorted \((n, T)\)-histograms would actually prove that this mechanism ensures the minimum average-error when it used to release integer partitions under differential privacy.
We address the problem of function release under differential privacy, by developing a functional mechanism, we term the Bernstein mechanism, that applies under the weak assumptions of oracle access to target function evaluation and sensitivity. We achieve this result by leveraging the iterated Bernstein operator for polynomial approximation of the target function, and polynomial coefficient perturbation. We establish fast convergence rates on the error introduced by the mechanism, and provide a lower bound on the utility achievable for any functional mechanism that is $\varepsilon$-differentially private.

This chapter is based on the paper “The Bernstein Mechanism: Function Release under Differential Privacy” [2], which is a joint work of the author with Benjamin I. P. Rubinstein. All results in this chapter were originally published in this paper, except for some additional remarks and proofs which only appear in the full report [1].

6.1 Introduction

As discussed in Section 3.3.1, the Laplace mechanism [30] is a de facto approach for converting vector-valued functions to differential privacy. In this chapter, we seek an equivalent approach for privatizing function-valued mappings. We achieve this goal through the development of a novel Bernstein functional mechanism. Unlike existing mechanisms, ours applies to releasing explicitly and implicitly defined functions (such as algorithmic black boxes), and is characterized by a full theoretical analysis.

The setting we consider is the release of functions that depend on privacy-sensitive training data, and that can be subsequently evaluated on arbitrary test points. This non-interactive setting matches a wide variety of learning
Chapter 6. Function Release and the Bernstein Mechanism

tasks from naive Bayes classification and non-parametric methods (kernel density estimation and regression) where the function of train and test data is explicit, to generalized linear models and support vector machines where the function is only implicitly defined by an iterative algorithm. Our generic mechanism is based on functional approximation by Bernstein basis polynomials, specifically via an iterated Bernstein operator. Privacy is guaranteed by sanitizing the coefficients of approximation, which requires only function evaluation. It is the very limited oracle access required by our mechanism—to non-private function evaluation and sensitivity—that grants it broad applicability akin to the Laplace mechanism.

The Bernstein polynomials central to our mechanism are used in the Stone-Weierstrass theorem to uniformly approximate any continuous function on a closed interval. Moreover, the Bernstein operator offers several advantages such as data-independent bounds, no requirement of access to target function derivatives, and yields approximations that are pointwise convex combinations of the function evaluations on a cover. As a result, applying privacy-preserving perturbations to the approximation’s coefficients permits us to control utility and achieve fast convergence rates.

In addition to being analyzed in full, the Bernstein mechanism is easy to use. We demonstrate this with a variety of example analyses of the mechanism applied to learners (see Section 6.6). Finally, we provide a lower bound that fundamentally limits utility under private function release, partly resolving a question posed by Hall et al. [42]. This matches (up to logarithmic factors) our upper bound in the linear case.

6.2 Setting

As usual, we consider \( \mathcal{X} \) an arbitrary (and possibly infinite) domain, and \( d \in \mathcal{X}^n \) a database of \( n \) points in \( \mathcal{X} \). For a positive integer \( \ell \), let \( \mathcal{Y} = [0, 1]^{\ell} \) be a set of query points and \( G : \mathcal{X}^n \times \mathcal{Y} \rightarrow \mathbb{R} \) the target function. Once the database \( d \) is fixed, we denote by \( G_d = G(d, \cdot) \) the function parameterized by \( d \) that we aim to release. For example: \( d \) might represent a training set—over \( \mathcal{X} \) a product space of feature vectors and labels—with \( \mathcal{Y} \) representing test points from the same feature space; \( G_d \) would then be a classifier resulting from training on \( d \). Section 6.6 presents some examples for \( G \). In Section 6.3, we show how to privately release the function \( G_d \) and we provide alternative error bounds depending on the regularity of \( G \).

**Definition 6.1.** Let \( H \) be a positive integer and \( T > 0 \). A function \( g : [0, 1]^\ell \rightarrow \mathbb{R} \) is \((H, T)\)-smooth if it is \( \mathcal{C}^H([0, 1]^\ell) \) and its partial derivatives up to order \( H \) are all bounded by \( T \).
6.3. The Bernstein Mechanism

Definition 6.2. Let $0 \leq \zeta \leq 1$ and $L > 0$. A function $g : [0, 1]^\ell \to \mathbb{R}$ is $(\zeta, L)$-Hölder continuous if, for every $x, y \in [0, 1]^\ell$, $|g(x) - g(y)| \leq L \|x - y\|_\zeta$. When $\zeta = 1$, we refer to $g$ as $L$-Lipschitz.

Our goal is to develop a private release mechanism for the function $G_d$ in the non-interactive setting. Such a non-interactive mechanism takes a function $G$ and a database $d$ as inputs and outputs a synopsis $O$ which can be used to evaluate the function $G_d$ on $Y$ without accessing the database $d$ further. Given a (differentially private) mechanism for the function $G_d$, we measure its accuracy as follows.

Definition 6.3. Let $G : \mathcal{X}^n \times \mathcal{Y} \to \mathbb{R}$. A mechanism $M$ is $(\alpha, \beta)$-accurate with respect to $G_d$ if, for any database $d \in \mathcal{X}^n$ and $O = M(d)$, with probability at least $1 - \beta$ over the randomness of $M$ the following holds:

$$\sup_{y \in \mathcal{Y}} |O(y) - G_d(y)| \leq \alpha.$$  

6.3 The Bernstein Mechanism

Algorithm 6.1 introduces a differentially-private mechanism for releasing $G_d : \mathcal{Y} \to \mathbb{R}$, a family of $(H, T)$-smooth or $(\zeta, L)$-Hölder continuous functions, parameterized by $d \in \mathcal{X}^n$.

Algorithm 6.1 The Bernstein mechanism

Sanitization

Inputs: private dataset $d \in \mathcal{X}^n$; sensitivity $\Delta(G)$; oracle access to target $G : \mathcal{X}^n \times \mathcal{Y} \to \mathbb{R}$

Parameters: cover size $k \in \mathbb{N}_+$; Bernstein order $H \in \mathbb{N}_+$; privacy parameter $\varepsilon > 0$

1: $\mathcal{L} \leftarrow \{0, 1/k, 2/k, \ldots, 1\}^\ell$ ▶ Lattice cover of $\mathcal{Y}$
2: $\lambda \leftarrow \Delta(G)(k + 1)H/\varepsilon$ ▶ Perturbation scale
3: For each $p = (p_1, \ldots, p_\ell) \in \mathcal{L}$:
4: $\widehat{G_d}(p) \leftarrow G_d(p) + Z$, where $Z \sim \text{Lap}(\lambda)$
5: Return: $\{\widehat{G_d}(p) : p \in \mathcal{L}\}$

Evaluation

Inputs: query $y \in \mathcal{Y}$; private response $\{\widehat{G_d}(p) : p \in \mathcal{L}\}$

6: $b_{\nu_1, \ldots, \nu_\ell}^{(H)} \leftarrow \text{Compute basis}$ ▶ See Equation (6.1)
7: Return: $\sum_{\nu_1 = 0}^k \sum_{\nu_2 = 0}^k \cdots \sum_{\nu_\ell = 0}^k \widehat{G_d}\left(\frac{\nu_1}{k}, \frac{\nu_2}{k}, \ldots, \frac{\nu_\ell}{k}\right) \prod_{i=1}^\ell b_{\nu_i, k}^{(H)}(y_i)$

The mechanism presented in Algorithm 6.1 makes use of the iterated Bernstein polynomial of $G_d$, which we introduce next. For a more comprehensive
survey we refer the reader to Lorentz [63], and Micchelli [67]. This approximation consists of a linear combination of so-called Bernstein basis polynomials, whose coefficients are evaluations of the target $G_d$ on a (lattice) cover $\mathcal{L}$. In what follows, we briefly introduce the univariate Bernstein basis polynomials and state some of their properties.

**Definition 6.4.** Let $k$ be a positive integer. The Bernstein basis polynomials of degree $k$ are defined as

$$b_{\nu,k}(y) = \binom{k}{\nu} y^\nu (1 - y)^{k-\nu},$$

for $\nu = 0, \ldots, k$.

**Proposition 6.1** (Lorentz [63]). For every $y \in [0,1]$, any positive integer $k$ and $0 \leq \nu \leq k$, the following holds:

$$b_{\nu,k}(y) \geq 0 \quad \text{and} \quad \sum_{\nu=0}^{k} b_{\nu,k}(y) = 1 .$$

In order to introduce the iterated Bernstein polynomial of a function, we first need to recall the Bernstein operator.

**Definition 6.5.** Let $g: [0,1] \to \mathbb{R}$ and $k$ be a positive integer. The Bernstein polynomial of $g$ of degree $k$ is defined as

$$B_k(g; y) = \sum_{\nu=0}^{k} g\left(\frac{\nu}{k}\right) b_{\nu,k}(y) .$$

The Bernstein operator $B_k$ maps a function $g$, defined on $[0,1]$, to $B_k g$, where the function $B_k g$ evaluated at $y$ is $B_k(g; y)$. Note that the Bernstein operator is linear and if $g(y) \in [a_1, a_2]$ for every $y \in [0,1]$, then from Proposition 6.1 it follows that $B_k(g; y) \in [a_1, a_2]$ for every positive integer $k$ and $y \in [0,1]$. Moreover, it is not hard to see that any linear function is a fixed point for $B_k$, as we briefly recall next.\(^7\) Let $k \geq 1$, and $g(y) = my + q$, for $m, q \in \mathbb{R}$ and $y \in [0,1]$. We thus have

$$B_k(g; y) = \sum_{\nu=0}^{k} g\left(\frac{\nu}{k}\right) b_{\nu,k}(y)$$

\[= \frac{m}{k} \sum_{\nu=0}^{k} \nu b_{\nu,k}(y) + q \sum_{\nu=0}^{k} b_{\nu,k}(y)\]

\[= my + q ,\]

\(^7\)This is a classical result, but we provide a short proof for completeness.
since $\sum_{\nu=0}^{k} b_{\nu,k}(y) = 1$ and $\sum_{\nu=0}^{k} \nu b_{\nu,k}(y) = ky$.

**Definition 6.6** (Micchelli [67]). Let $H, k$ be positive integers. The iterated Bernstein operator of order $H$ (and degree $k$) is defined as the sequence of linear operators

$$B_k^{(H)} = I - (I - B_k)^H = \sum_{i=1}^{H} \binom{H}{i} (-1)^{i-1} B_k^i ,$$

where $B_k^0 = I$ denotes the identity operator and $B_k^i = B_k \circ B_k^{i-1}$ for $i \geq 1$. The iterated Bernstein polynomial of order $H$ (and degree $k$) can then be computed as follows:

$$B_k^{(H)}(g; y) = \sum_{\nu=0}^{k} g \left( \frac{\nu}{k} \right) b_{\nu,k}^{(H)}(y) ,$$

where

$$b_{\nu,k}^{(H)}(y) = \sum_{i=1}^{H} \binom{H}{i} (-1)^{i-1} B_k^{i-1}(b_{\nu,k}; y) . \quad (6.1)$$

Note that $B_k^{(1)} = B_k$. Although the bases $b_{\nu,k}^{(H)}$ are not always positive for $H \geq 2$, we still have $\sum_{\nu=0}^{k} b_{\nu,k}^{(H)}(y) = 1$ for every $y \in [0,1]$. The iterated Bernstein polynomial of a multivariate function $g: [0,1]^\ell \to \mathbb{R}$ is analogously defined.

**Definition 6.7.** Assume $g: [0,1]^\ell \to \mathbb{R}$ and let $k_1, \ldots, k_\ell, H$ be positive integers. The (multivariate) iterated Bernstein polynomial of $g$ (of order $H$ and degree $k$) is defined as

$$B_{k_1,\ldots,k_\ell}^{(H)}(g; y) = \sum_{\nu_1=0}^{k_1} \sum_{\nu_2=0}^{k_2} \cdots \sum_{\nu_\ell=0}^{k_\ell} g \left( \frac{\nu_1}{k_1}, \frac{\nu_2}{k_2}, \ldots, \frac{\nu_\ell}{k_\ell} \right) \prod_{i=1}^{\ell} b_{\nu_i,k_i}^{(H)}(y_i) .$$

For the sake of clarity, we fix a user-selected $k \in \mathbb{N}_+$ such that $k_1 = \ldots = k_\ell = k$. The Bernstein mechanism perturbs the evaluation of $G_d$ on a lattice cover $\mathfrak{L}$ of $\mathcal{Y} = [0,1]^\ell$ parameterized by $k$.

### 6.4 Privacy and Utility Analysis

In this section, we will prove one of the main results of this chapter, namely Theorem 6.2 below. We assume $\ell$ to be an arbitrary but fixed constant with $\mathcal{Y} = [0,1]^\ell$. We underline that this is a common assumption in the differential privacy literature, especially when dealing with Euclidean spaces [14, 27, 86, 62, 85].
Theorem 6.2. Let $\ell, H \in \mathbb{N}_+, 0 < \zeta \leq 1, L > 0$ and $T > 0$ be constants. Let $X$ be an arbitrary space and $Y = [0, 1]^\ell$. Let furthermore $G : X^n \times Y \to \mathbb{R}$ with $\Delta(G) = o(1)$. For $\varepsilon > 0$, the Bernstein mechanism $M$ described in Algorithm 6.1 provides $\varepsilon$-differential privacy. Moreover, for $0 < \beta < 1$ the mechanism $M$ is $(\alpha, \beta)$-accurate with error scaling as follows, where hidden constants and logarithmic factors depend on $\ell, L, \zeta, T, H$ and $\beta$.

(i) If $G_d$ is $(2H, T)$-smooth for every $d \in X^n$, there exists $k = k(\Delta(G), \varepsilon, \ell, H, T)$ such that $\alpha = \tilde{O}\left(\frac{\Delta(G)}{\varepsilon}\right)^{\frac{H}{\ell}}$;

(ii) If $G_d$ is $(\zeta, L)$-Hölder continuous for every $d \in X^n$, there exists $k = k(\Delta(G), \varepsilon, \ell, \zeta, L)$ such that $\alpha = \tilde{O}\left(\frac{\Delta(G)}{\varepsilon}\right)^{\frac{\zeta}{2\ell}}$; and

(iii) If $G_d$ is linear for every $d \in X^n$, there exists a constant $k$ such that $\alpha = \tilde{O}\left(\frac{\Delta(G)}{\varepsilon}\right)$.

Moreover, if $1/\Delta(G) \leq \text{poly}(n)$, then the running-time of the mechanism and the running-time per evaluation are both polynomial in $n$ and $1/\varepsilon$.

For the sake of clarity, in Section 6.4.1 we provide a self-contained proof of Theorem 6.2 for $\ell = 1$. Although this is not a prerequisite to the general result, it reflects the building blocks used in the general proof, provided in Section 6.4.2.

6.4.1 Proof of Theorem 6.2 for $\ell = 1$

Let us fix $k$, a positive integer. To prove privacy we note that only the coefficients of the Bernstein polynomial of $G_d$ are sensitive and need to be protected. In order to provide $\varepsilon$-differential privacy, these coefficients—evaluations of target $G_d$ on a cover—are perturbed by means of Lemma 3.4. In this way, we can release the sanitized coefficients and use them for unlimited, efficient evaluation of the approximation of $G_d$ over $Y = [0, 1]$, without further access to the dataset $d$.

Lemma 6.3. Let $\varepsilon > 0$. Then the Bernstein mechanism $M$ (with parameter $\varepsilon$) provides $\varepsilon$-differential privacy.

Proof. Let $d' \in X^n$ be a second database differing from $d$ in one entry only. Let furthermore $g : X^n \to \mathbb{R}^{k+1}$ be the map defined by

$$g(d) = \left(G_d\left(\frac{0}{k}\right), G_d\left(\frac{1}{k}\right), \ldots, G_d\left(\frac{k}{k}\right)\right).$$

Then

$$\Delta(g) = \sup_{d \approx d'} \|g(d) - g(d')\|_1$$
6.4. Privacy and Utility Analysis

\[ \leq \sum_{\nu=0}^{k} \sup_{d \approx d'} \left| G_d \left( \frac{\nu}{k} \right) - G_{d'} \left( \frac{\nu}{k} \right) \right| \leq \Delta(G)(k + 1). \]

According to Lemma 3.4 (applied with \( k + 1 \) in place of \( r \)), the mechanism \( \mathcal{M} \) provides \( \varepsilon \)-differential privacy.

In order to analyze the accuracy of the Bernstein mechanism, we denote by

\[ \hat{B}_k^{(H)}(G_d; y) = \sum_{\nu=0}^{k} \left[ G_d \left( \frac{\nu}{k} \right) + Z_\nu \right] b_{\nu,k}^{(H)}(y) \]

the iterated Bernstein polynomial of order \( H \) constructed using the coefficients output by the mechanism \( \mathcal{M} \). We remind the reader that, for \( \nu = 0, \ldots, k \), \( Z_\nu \sim \text{Lap}(\lambda) \), where \( \lambda = \Delta(G)(k + 1)/\varepsilon \). The error \( \alpha \) introduced by the mechanism can be expressed as follows:

\[ \alpha = \max_{y \in [0, 1]} \left| G_d(y) - \hat{B}_k^{(H)}(G_d; y) \right| \]

\[ \leq \max_{y \in [0, 1]} \left| \hat{B}_k^{(H)}(G_d; y) - B_k^{(H)}(G_d; y) \right| + \max_{y \in [0, 1]} \left| G_d(y) - B_k^{(H)}(G_d; y) \right|. \] (6.3)

For every \( y \in [0, 1] \), the first summand in Equation (6.3) consists of the absolute value of an affine combination of independent Laplace-distributed random variables. In order to bound this sum, we use the following tail bound.

**Proposition 6.4.** Let \( Z_0, \ldots, Z_k \sim \text{i.i.d. Lap}(\lambda) \), and \( \xi \geq 0 \). Then, the following holds:

\[ \Pr \left[ \max_{y \in [0, 1]} \left| \sum_{\nu=0}^{k} Z_\nu b_{\nu,k}^{(H)}(y) \right| \geq \xi \right] \leq (k + 1) \exp \left( -\frac{\xi}{(2^H - 1)\lambda} \right). \]

**Proof.** For the sake of simplicity, let

\[ Z = (Z_0, \ldots, Z_k) \quad \text{and} \quad b(y) = (b_{0,k}^{(H)}(y), \ldots, b_{k,k}^{(H)}(y)). \]

For \( y \in [0, 1] \), the \( L_1 \)-norm of \( b(y) \) can be bounded as follows:

\[ \|b(y)\|_1 = \sum_{\nu=0}^{k} \left| \sum_{i=1}^{H} \binom{H}{i} (-1)^{i-1} B_k^{i-1}(b_{\nu,k}; y) \right| \leq \sum_{\nu=0}^{k} \left| \sum_{i=1}^{H} \binom{H}{i} B_k^{i-1}(b_{\nu,k}; y) \right|. \] (6.4)
Equality (6.6) follows from the fact that the Bernstein operator is a monotone operator (see for example Phillips [69]) and the basis polynomials $b_{ν,k}(y) ∈ [0, 1]$ for $y ∈ [0, 1]$. Equality (6.9) is an immediate consequence of Proposition 6.1. We thus have

$$\Pr \left[ \max_{y ∈ [0, 1]} \left| \sum_{ν=0}^{k} Z_ν b^{(H)}_ν (y) \right| ≥ ξ \right] = \Pr \left[ \max_{y ∈ [0, 1]} |⟨Z, b(y)⟩| ≥ ξ \right]$$

$$≤ \Pr \left[ \max_{y ∈ [0, 1]} \|Z\|_∞ \|b(y)\|_1 ≥ ξ \right] \quad \text{(6.12)}$$

$$≤ \Pr \left[ (2^H - 1) \|Z\|_∞ ≥ ξ \right] \quad \text{(6.13)}$$

where (6.12) and (6.13) follow from Hölder’s inequality and Inequality (6.10), respectively. We remind the reader that, if $Z ∼ \text{Lap}(λ)$, $\Pr[|Z| ≥ ξ] ≤ \exp(-ξ/λ)$. Therefore, by the union bound,

$$\Pr \left[ (2^H - 1) \|Z\|_∞ ≥ ξ \right] = \Pr \left[ \|Z\|_∞ ≥ \frac{ξ}{(2^H - 1)} \right]$$

$$≤ (k + 1) \exp \left( -\frac{ξ}{(2^H - 1)λ} \right) ,$$

concluding the proof.

Proposition 6.4 implies that with probability at least $1 - β$ the first summand in (6.3) is bounded by $O(Δ(G)k \log(k/β)/ε)$. In order to bound the second summand we make use of the following convergence rates.

**Theorem 6.5** (Micchelli [67]). Let $H$ be a positive integer and $T > 0$. If $g: [0, 1] → \mathbb{R}$ is a $(2H, T)$-smooth function, then, for all positive integers $k$ and $y ∈ [0, 1],

$$\left| g(y) - B_k^{(H)}(g; y) \right| ≤ TD_H k^{-H} ,$$

where $D_H$ is a constant independent of $k, g$ and $y ∈ [0, 1].$
6.4. Privacy and Utility Analysis

**Theorem 6.6** ([51][64]). Let $0 < \zeta \leq 1$ and $L > 0$. If $g: [0, 1] \to \mathbb{R}$ is a $(\zeta, L)$-Hölder continuous function, then

$$|g(y) - B_k^{(1)}(g; y)| \leq L(4k)^{-\zeta/2},$$

for all positive integers $k$ and $y \in [0, 1]$.

According to the regularity of $G_d$, the second summand in (6.3) can be bounded by a decreasing function $g(k)$. All in all, the error in (6.2) can be bounded as follows:

$$\alpha = O\left(g(k) + \frac{\Delta(G)k}{\epsilon} \log(k/\beta)\right).$$  \hspace{1cm} (6.14)

Since the second summand in Equation (6.14) is an increasing function in $k$, the optimal value for $k$ (up to logarithmic factors) is achieved when $k$ satisfies

$$g(k) = \frac{\Delta(G)k}{\epsilon}. \hspace{1cm} (6.15)$$

Solving Equation (6.15) with the bounds for $g(k)$ provided in Theorems 6.5 and 6.6, and substituting the thus obtained value of $k$ into (6.14) yield the first two statements of Theorem 6.2. The bound when $G_d$ is linear follows from the fact that, for $H = 1$ and $k = 1$, the approximation error in (6.3) is zero. The error is thus bounded by $\tilde{O}(\Delta(G)/\epsilon)$. The running time of the mechanism and the running time for answering a query is linear in $k$ and hence upper bounded by a polynomial in $\log(1/\epsilon)$.

6.4.2 Proof of Theorem 6.2

The proof of Theorem 6.2 for $\ell > 1$ closely follows the workflow we used to demonstrate the statement for the one-dimensional case ($\ell = 1$) in Section 6.4.1. As described in Algorithm 6.1, the Bernstein mechanism perturbs the evaluation of the function $G_d$ on a cover of $\mathcal{Y} = [0, 1]^\ell$. Since these evaluations are perturbed by means of Lemma 3.4, differential privacy is immediately established.

In order to analyze the accuracy of the mechanism, we denote by $\hat{B}_k^{(H)}(G_d; y)$ the iterated Bernstein polynomial of order $H$ constructed using the coefficients output by the mechanism $\mathcal{M}$. The error $\alpha$ introduced by the mechanism can be expressed as follows:

$$\alpha = \max_{y \in [0, 1]^\ell} \left|G_d(y) - \hat{B}_k^{(H)}(G_d; y)\right|$$  \hspace{1cm} (6.16)
\[
\leq \max_{y \in [0,1]^\ell} \left| B_k^{(H)}(G_d; \mathbf{y}) - B_k^{(H)}(G_d; \mathbf{y}) \right| + \max_{y \in [0,1]^\ell} \left| G_d(y) - B_k^{(H)}(G_d; \mathbf{y}) \right|.
\]

\quad (6.17)

For every \( y \in [0,1]^\ell \), the first summand in (6.17) consists of the absolute value of an affine combination of independent Laplace-distributed random variables.

**Proposition 6.7.** Let \( \Gamma = \{ \nu \in \mathbb{N}^\ell : 0 \leq \nu_j \leq k \text{ for } 1 \leq j \leq \ell \} \). For every \( \nu = (\nu_1, \ldots, \nu_\ell) \in \Gamma \) let \( Z_\nu \overset{\text{i.i.d.}}{\sim} \text{Lap}(\lambda) \). Then, for any \( \xi \geq 0 \),

\[
\Pr \left[ \max_{y \in [0,1]^\ell} \left| \sum_{\nu_1=0}^k \sum_{\nu_2=0}^k \cdots \sum_{\nu_\ell=0}^k Z_\nu \prod_{i=1}^\ell b_{\nu_i,k}^{(H)}(y_i) \right| \geq \xi \right] \leq (k+1)^\ell \exp \left( -\frac{\xi}{(2^H - 1)^\ell \lambda} \right).
\]

**Proof.** The statement of Proposition 6.7 follows from the argument used in the proof of Proposition 6.4, with some minor changes. Let

\[
Z = (Z_\nu)_{\nu \in \Gamma} \quad \text{and} \quad \mathbf{b}(y) = \left( \prod_{i=1}^\ell b_{\nu_i,k}^{(H)}(y_i) \right)_{\nu \in \Gamma}.
\]

For every \( y \in [0,1]^\ell \), we have

\[
\|\mathbf{b}(y)\|_1 = \sum_{\nu_1=0}^k \sum_{\nu_2=0}^k \cdots \sum_{\nu_\ell=0}^k \prod_{i=1}^\ell \left| b_{\nu_i,k}^{(H)}(y_i) \right| = \left( \sum_{\nu_1=0}^k \cdots \sum_{\nu_\ell=0}^k \prod_{i=1}^\ell \left| b_{\nu_i,k}^{(H)}(y_i) \right| \right) \sum_{\nu_1=0}^k \left| b_{\nu_1,k}^{(H)}(y_1) \right| \leq (2^H - 1)^\ell,
\]

since, according to Equation (6.10), \( \sum_{\nu_1=0}^k \left| b_{\nu_1,k}^{(H)}(y_j) \right| \leq (2^H - 1) \) for every \( j \in \{1, \ldots, \ell\} \). Similarly to Proposition 6.4, we have

\[
\Pr \left[ \max_{y \in [0,1]^\ell} \left| \sum_{\nu_1=0}^k \sum_{\nu_2=0}^k \cdots \sum_{\nu_\ell=0}^k Z_\nu \prod_{i=1}^\ell b_{\nu_i,k}^{(H)}(y_i) \right| \geq \xi \right] = \Pr \left[ \max_{y \in [0,1]^\ell} \left| \langle Z, \mathbf{b}(y) \rangle \right| \geq \xi \right]
\]

\[
\leq \Pr \left[ \max_{y \in [0,1]^\ell} \| Z \|_\infty \| \mathbf{b}(y) \|_1 \geq \xi \right]
\]

\[
\leq \Pr \left[ (2^H - 1)^\ell \| Z \|_\infty \geq \xi \right]
\]

\[
\leq (k+1)^\ell \exp \left( -\frac{\xi}{(2^H - 1)^\ell \lambda} \right),
\]

where the last inequality follows from the application of the union bound. \( \square \)
Proposition 6.7 implies that with probability at least $1 - \beta$ the first summand in Equation (6.17) is bounded by $O\left(\Delta(G)k^\ell \log(k^\ell/\beta/\varepsilon)\right)$. In order to bound the second summand we make use of the (unidimensional) convergence rates provided in Theorems 6.5 and 6.6. By induction, it is possible to show that the approximation error of the multivariate iterated Bernstein polynomial is bounded by $O\left(\ell g(k)\right) = O(g(k))$ (we remind the reader that $\ell$ is assumed to be constant in this chapter), if the error of the corresponding univariate polynomial is bounded by $g(k)$. For the sake of completeness, we briefly prove this result in the following paragraph.

Approximation Error of Multivariate Bernstein Polynomials. For simplicity, we assume that $g: [0, 1]^\ell \to \mathbb{R}$ is a $(\zeta, L)$-Hölder continuous function. The proof for $(H, T)$-smooth functions follows from a similar argument, with minor changes. The argument we present here is by induction on $\ell$. The base case ($\ell = 1$) follows from the fact that the Bernstein polynomial $B_k(g; y)$ converges uniformly to $g$ in the interval $[0, 1]$, as shown in Theorem 6.6. Assume now

$$|B_k(g; y) - g(y)| \leq \ell L \left(\frac{1}{4k}\right)^{\zeta/2},$$

for every $y \in [0, 1]^\ell$. Let $g: [0, 1]^{\ell+1} \to \mathbb{R}$ be a $(\zeta, L)$-Hölder continuous function and let $B_k(g; y)$ be the corresponding Bernstein polynomial. For every $y = (y_1, \ldots, y_{\ell+1}) \in [0, 1]^{\ell+1}$, let

$$B'(g; y) = \sum_{\nu_1=0}^k \sum_{\nu_2=0}^k \cdots \sum_{\nu_\ell=0}^k g\left(\frac{\nu_1}{k}, \frac{\nu_2}{k}, \ldots, \frac{\nu_\ell}{k}, y_{\ell+1}\right) \prod_{i=1}^\ell b_{\nu_i,k}(y_i).$$

The error $|B_k(g; y) - g(y)|$ can then be bounded by

$$|B_k(g; y) - g(y)| \leq |B_k(g; y) - B'(g; y)| + |B'(g; y) - g(y)| \leq L \left(\frac{1}{4k}\right)^{\zeta/2} + \ell L \left(\frac{1}{4k}\right)^{\zeta/2} = (\ell + 1) L \left(\frac{1}{4k}\right)^{\zeta/2}.$$ 

In fact, the second term of (6.18) is the error of the Bernstein polynomial of $g$ seen as a function of $y_1, \ldots, y_\ell$ only. The corresponding bound then follows from the inductive step. On the other hand, the first summand corresponds to the approximation error of the (univariate) Bernstein polynomial of $B'(g, y)$ as a function of the remaining variable $y_{\ell+1}$. The statement for $(H, T)$-smooth functions is similarly obtained by replacing $B_k$ with $B_k(H)$, and using the bound of Theorem 6.5.

\[\square\]
We can now continue with the proof of Theorem 6.2. By virtue of the discussion above, the error $\alpha$ introduced by the mechanism can be bounded by

$$\alpha = O \left( g(k) + \frac{\Delta(G)k^\ell}{\varepsilon} \log(k^\ell / \beta) \right).$$  \hspace{1cm} (6.19)$$

Since $g(k)$ is a decreasing function in $k$ and the second summand in (6.19) is an increasing function in $k$, the optimal value for $k$ (up to logarithmic factors) is achieved when $k$ satisfies

$$g(k) = \frac{\Delta(G)k^\ell}{\varepsilon}. \hspace{1cm} (6.20)$$

Solving Equation (6.20) with the bound for $g(k)$ provided in Theorem 6.5 yields

$$k = \max \left\{ 1, \left( \frac{\varepsilon}{\Delta(G)} \right)^{\frac{1}{\pi+\tau}} \right\},$$

and substituting the thus obtained value of $k$ into Equation (6.19) yields the first statement of Theorem 6.2. Similarly, using the bound for $g(k)$ provided in Theorem 6.6 we get the result for Hölder continuous functions. The bound for linear functions follows from the fact that the approximation error is zero for $H = 1$ and $k = 1$, since linear functions are fixed points of $B_1^{(1)}$. Finally, the analysis of the running time follows from observing that, for the cover size $k$ we computed,

$$k^\ell \leq \max \left\{ 1, \frac{\varepsilon}{\Delta(G)} \right\},$$

and thus by $\text{poly}(n)$ if $1/\Delta(G) \leq \text{poly}(n)$.

### 6.4.3 Discussion

**Comparison to Baseline.** Algorithm 6.1 is based on a relatively simple approach: it evaluates the target function on a lattice cover, adding Laplace noise for privacy. One might be tempted to approximate the input function by rounding a query point $y$ to the nearest lattice point $p \in \mathcal{L}$ and releasing the corresponding noisy evaluation $\hat{G}_d(p)$. Although it is straightforward to prove that, for $(\zeta, L)$-Hölder continuous functions, such a piecewise constant approximation achieves error $O(1/k^\zeta)$, this upper bound is essentially tight, as it can be shown by considering the approximation error it achieves for linear functions. Therefore, this method has two main disadvantages: the output function is not even continuous (although we always consider continuous input functions) and for highly smooth input functions it cannot achieve the fast
6.5. Lower Bound

convergence rates of the Bernstein mechanism. In Section 6.6, we offer further examples supporting this argument.

\((\varepsilon, \delta)\)-Differential Privacy. Note that our analysis can be easily extended to the relaxed notion of approximate differential privacy using advanced composition theorems (see for example Dwork and Roth [28]) instead of sequential composition (introduced in Proposition 3.3). Specifically, it suffices to choose the perturbation scale

\[
\lambda_{\delta} := \frac{2\Delta(G)\sqrt{2(k + 1)\ell \log(1/\delta)}}{\varepsilon}.
\]

Theorem 6.8. Let \(0 < \delta < 1\). Under the same assumptions of Theorem 6.2, the Bernstein mechanism \(\mathcal{M}\) (with perturbation scale \(\lambda_{\delta}\)) provides \((\varepsilon, \delta)\)-differential privacy and is \((\alpha, \beta)\)-accurate with error scaling as follows.

(i) If \(G_d\) is \((2H,T)\)-smooth for every \(d \in \mathcal{X}^n\), there exists \(k = k(\Delta(G), \varepsilon, \ell, H, T)\) such that \(\alpha = \tilde{O}\left(\frac{\Delta(G)}{\varepsilon}^{2H}\right)\); and

(ii) If \(G_d\) is \((\zeta,L)\)-Hölder continuous for every \(d \in \mathcal{X}^n\), there exists \(k = k(\Delta(G), \varepsilon, \ell, \zeta, L)\) such that \(\alpha = \tilde{O}\left(\frac{\Delta(G)}{\varepsilon}^{\frac{\zeta}{L}}\right)\).

Even though this relaxation allows for improved accuracy, in this chapter we explore a different point on the privacy-utility Pareto front and focus our attention on \(\varepsilon\)-differential privacy, since there generally is a significant motivation for achieving stronger privacy guarantees. Moreover, to the best of our knowledge, it is unknown whether previous solutions [42] even apply to this framework.

6.5 Lower Bound

In this section we present a lower bound on the error that any \(\varepsilon\)-differentially private mechanism approximating a function \(G: \mathcal{X}^n \times \mathcal{Y} \to \mathbb{R}\) must introduce.

Theorem 6.9. Let \(\varepsilon > 0\). Then, there exists a function \(G: \mathcal{X}^n \times \mathcal{Y} \to \mathbb{R}\) such that the \(L_{\infty}\)-error incurred by any \(\varepsilon\)-differentially private mechanism approximating \(G\) is \(\Omega\left(\frac{\Delta(G)}{\varepsilon}\right)\) with arbitrarily high probability.

Proof. In order to prove Theorem 6.9, we consider \(\mathcal{X} \subset [0,1]^\ell\) to be a finite set and, without loss of generality, we view the database \(d\) as an element of \(\mathcal{X}^n\) or as an element of \(\mathbb{N}^{|\mathcal{X}|}\), i.e., a histogram over the elements of \(\mathcal{X}\), interchangeably.\(^8\)

We can then make use of a general result due to De [23].

\(^8\)We refer the reader to Section 4.2 for a more thorough explanation of the histogram representation.
Proposition 6.10 (De [23]). Consider \(d^{[1]}, d^{[2]}, \ldots, d^{[2^s]} \in \mathbb{N}^{[\mathcal{X}]}\) such that, for every \(i = 1, \ldots, 2^s\), \(\|d^{[i]}\|_1 \leq n\) and, for \(i \neq j\), \(\|d^{[i]} - d^{[j]}\|_1 \leq (s - 1)/\varepsilon\). Moreover, let \(g: \mathbb{N}^{[\mathcal{X}]} \rightarrow \mathbb{R}^\ast\) be such that, for any \(i \neq j\), \(\|g(d^{[i]}) - g(d^{[j]})\|_\infty \geq \eta\). Then, any mechanism which is \(\varepsilon\)-differentially private for the query \(g\) on databases of size \(n\) must introduce an \(L_\infty\)-error which is \(\Omega(\eta)\) with arbitrarily high probability.

Therefore, we only need to show that there exists a suitable sequence of databases \(d^{[1]}, d^{[2]}, \ldots, d^{[2^s]}\), a function \(G: \mathcal{X}^n \times \mathcal{Y} \rightarrow \mathbb{R}\), and a \(y \in \mathcal{Y}\) such that \(G(\cdot, y)\) satisfies the assumptions of Proposition 6.10. We actually show that this holds for every \(y \in \mathcal{Y}\). Let \(\varepsilon > 0\) and \(V\) be a non-negative integer. We define \(\mathcal{X} = \{0, 1/(V+8), 2/(V+8), \ldots, 1\}\). Note that \(M = |\mathcal{X}| = (V+9)^\ell\). Let furthermore \(c = \lceil 1/\varepsilon \rceil\) and \(n = V + c\). The function \(G: \mathcal{X}^n \times [0, 1]^\ell \rightarrow \mathbb{R}\) we consider is defined as follows:

\[
G(d, y) = \eta(n_0 + \ldots + n_{M-7} + 2n_{M-6} + \ldots + 8n_M + (y, 1)),
\]

where \(n_i\) corresponds to the number of entries in \(d\) whose value is \(x_i \in \mathcal{X}\), and \(1\) is the all-ones vector. For \(s = 3\), we consider the sequence of databases \(d^{[1]}, d^{[2]}, \ldots, d^{[8]}\), where, for \(j \in \{1, 2, \ldots, 8\}\), we have

\[
n^{[j]}_i = \begin{cases} 
1 & \text{for } i \in \{0, 1, \ldots, V - 1\} \\
c & \text{for } i = M - j + 8 \\
0 & \text{otherwise}
\end{cases}
\]

We first observe that, for every \(i \in \{1, 2, \ldots, 8\}\), \(\|d^{[i]}\|_1 = n\). Moreover, for \(i \neq j\), \(\|d^{[i]} - d^{[j]}\|_1 = 2c \leq 2/\varepsilon\). Finally, for \(i \neq j\), \(|G(d^{[i]}, y) - G(d^{[j]}, y)| \geq c\eta\) for every \(y \in [0, 1]^\ell\). Since \(\Delta(G) = 7\eta\), Proposition 6.10 implies that, with arbitrarily high probability, any \(\varepsilon\)-differentially private mechanism approximating \(G\) must introduce an \(L_\infty\)-error of order \(\Omega(\Delta(G)/\varepsilon)\).

### 6.6 Examples

In this section, we demonstrate the versatility of the Bernstein mechanism through the analysis of a range of example learners. The error bounds we provide are direct applications of Theorem 6.2.

**Kernel Density Estimation.** Let \(\mathcal{X} = [0, 1]^{\ell}\), and \(d \in \mathcal{X}^n\). In order to emphasize that, for \(i = 1, \ldots, n\), \(d_i\) is an \(\ell\)-dimensional vector, we define \(x^{[i]} := d_i\). For a given kernel\(^9\) \(\text{KN}_\Sigma\) with bandwidth \(\Sigma\) (a symmetric and

---

\(^9\)For a formal definition, we refer the interested reader to the book of Silverman [77].
positive definite $\ell \times \ell$ matrix), the kernel density estimator $G_{\Sigma}: \mathcal{X}^n \times \mathcal{Y} \rightarrow \mathbb{R}$ is defined as

$$G_{\Sigma}(d, y) = \frac{1}{n} \sum_{i=1}^{n} K_{\Sigma}(y - x[i]).$$

It is easy to see that

$$\Delta(G_{\Sigma}) \leq \sup_{y \in [-1, 1]^\ell} \frac{K_{\Sigma}(y)}{n} \cdot$$

For instance, if $K_{\Sigma}$ is the Gaussian kernel with covariance matrix $\Sigma$, then

$$\Delta(G_{\Sigma}) \leq \frac{1}{n \sqrt{(2\pi)^\ell \det(\Sigma)}}. \tag{6.21}$$

Moreover, observe that $G_{\Sigma}(d, \cdot)$ is an $(H, T)$-smooth function for any positive integer $H$. Hence the error introduced by the mechanism is

$$\tilde{O}\left( \frac{1}{n \varepsilon \sqrt{\det(\Sigma)}} \right)^{\frac{n}{177}},$$

with probability at least $1 - \beta$.

In Figure 6.1 we display the utility (averaged over 1000 repeats) of the Bernstein mechanism ($k = 20$) on 5000 points drawn from a mixture of two normal distributions $\mathcal{N}(0.5, 0.02)$ and $\mathcal{N}(0.75, 0.005)$ with weights 0.4 and 0.6,
respectively. We first observe that for every privacy parameter \( \varepsilon \) there is a suitable choice of \( H \) such that our mechanism always achieves better utility compared to the baseline (cf. Section 6.4.3). Moreover, accuracy improves for increasing \( H \), except for sufficiently large perturbations (corresponding to small \( \varepsilon \)) which more significantly affect higher-order basis functions (larger \( H \)). Private cross validation \([21, 20]\) can be used to tune \( H \). We conclude noting that the same error bounds can be provided by the mechanism of Wang et al. \([85]\), since the function \( G_\Sigma(d, \cdot) \) is separable in the training set \( d \), i.e.,

\[
G_\Sigma(d, \cdot) = \sum_{i=1}^{n} g_\Sigma(x_i, \cdot)
\]

However, this assumption is overly restrictive for many applications. In the following, we discuss how the Bernstein mechanism can be successfully applied to several such cases.

**Priestley-Chao Kernel Regression.** For ease of exposition, consider \( \ell = 1 \). For a constant \( a > 0 \), let \( X = [0, 1] \times [-a, a] \) and \( Y = [0, 1] \). Without loss of generality, consider datasets \( d \in X^n \) such that, for \( i \in \{1, \ldots, n\} \), \( d_i = (x_i, l_i) \), and \( x_1 \leq x_2 \leq \ldots \leq x_n \). Moreover, assume that, for every \( i \in \{1, \ldots, n\} \), there exists \( j \neq i \) such that \( |x_i - x_j| \leq c/n \), for a given (and publicly known) \( 0 < c \leq n \). Small values of \( c \) restrict the data space under consideration, whereas \( c = n \) corresponds to the general case \( d \in X^n \). For a kernel \( K \) and a bandwidth \( \sigma > 0 \), the Priestley-Chao kernel estimator \([71, 10]\) is defined as

\[
G_\sigma(d, y) = \frac{1}{\sigma} \sum_{i=2}^{n} (x_i - x_{i-1}) K\left(\frac{y - x_i}{\sigma}\right) l_i.
\]

This function is not separable in \( d \) and

\[
\Delta(G_\sigma) = \sup_{y \in Y} \Delta(G_\sigma(\cdot, y)) \leq \frac{4ac}{n\sigma} \sup_{y \in [-1,1]} K\left(\frac{y}{\sigma}\right).
\]

If \( K \) is the Gaussian kernel, then, with probability at least \( 1 - \beta \), the error introduced by the Bernstein mechanism can be bounded by

\[
\mathcal{O}\left(\frac{c}{n\sigma}\right)^{\frac{n}{1+\beta}}.
\]

**Naive Bayes Classification.** In this example we apply the Bernstein mechanism to a probabilistic learner. Without loss of generality, assume \( X' = [0, 1]^\ell \), \( X = X' \times \{-1, 1\} \), \( Y = X' \), and \( d \in X^n \), where, for \( i \in \{1, \ldots, n\} \), \( d_i = (x_i, l_i) \).
A naive Bayes classifier can be interpreted as $G : \mathcal{X}^n \times \mathcal{Y} \rightarrow \mathbb{R}$ such that

$$G_d(y) = \Pr(y|1,d) \Pr(1|d) - \Pr(y|-1,d) \Pr(-1|d) .$$

Predictions can then be made by assigning the instance $y$ to the class $1$ (resp. $-1$) if $G_d(y) \geq 0$ (resp. $G_d(y) < 0$). Since, for a class $l$, $\Pr(y|l,d) = \prod_{i=1}^{\ell} \Pr(y_i|l,d)$, it is easy to show that $G_d(\cdot)$ is an $(H,T)$-smooth function whenever each likelihood is estimated using a Gaussian distribution or KDE\[49\] (with a sufficiently smooth kernel). In the latter case, using a Gaussian kernel, the sensitivity of $G$ can be bounded as follows:

$$\Delta(G) = \sup_{y \in [0,1]^n, d \approx d'} |G(d,y) - G(d',y)|$$

$$\leq 2 \cdot \sup_{l \in \{-1,1\}, y \in [0,1]^n} |\Pr(y|l,d) \Pr(l|d) - \Pr(y|l,d') \Pr(l|d')| .$$

We assume that a class probability $\Pr(l|d)$ is estimated using the corresponding relative frequency in the training set $d$. Therefore, for $d \approx d'$, $\Pr(l|d) \leq \Pr(l|d') + 1/n$. Assume now that for every $y \in [0,1]^\ell$, $d \approx d' \in \mathcal{X}^n$, $l \in \{-1,1\}$, and $i \in \{1,\ldots,\ell\}$ there exists $0 \leq \xi < 1$ such that

$$|\Pr(y_i|l,d) - \Pr(y_i|l,d')| \leq \xi$$

holds. We then have

$$\Pr(y|l,d) = \prod_{i=1}^{\ell} \Pr(y_i|l,d)$$

$$\leq \prod_{i=1}^{\ell} (\Pr(y_i|l,d') + \xi)$$

$$\leq \prod_{i=1}^{\ell} \Pr(y_i|l,d') + (2^\ell - 1) \xi ,$$

where the last inequality follows from the fact that there are $2^\ell - 1$ cross products and each one of them has at least a $\xi$ factor. If each (unidimensional) likelihood is estimated using KDE\[49\] with a Gaussian kernel with bandwidth $\sigma$, $\xi$ corresponds to the upper bound on the sensitivity of KDE shown in (6.21). Putting all the pieces together, we obtain

$$\Delta(G) \leq 2 \cdot \sup_{l \in \{-1,1\}, y \in [0,1]^\ell, d \approx d'} \frac{2^\ell - 1}{n \sqrt{2\pi \sigma}} \cdot \Pr(l|d') + \frac{1}{n} \cdot \Pr(y|l,d)$$

$$\leq 2 \left( \frac{2^\ell - 1}{n \sqrt{2\pi \sigma}} + \frac{1}{n} \right) .$$
Chapter 6. Function Release and the Bernstein Mechanism

The error introduced by the Bernstein mechanism is thus bounded by

\[ \tilde{O}\left(\frac{1}{n^\sigma}\right)^{\frac{n}{n^\sigma}}, \]

with probability at least \(1 - \beta\).

**Regularized Empirical Risk Minimization.** In the next examples, the functions we aim to release are implicitly defined by an algorithm. Let \(X' = [0, 1]^d, X = X' \times [0, 1]\) and \(Y = X'\). Let \(LS\) be a convex and locally \(T'\)-Lipschitz (in the first argument) loss function. Let \(d \in X^n\), where, for \(i \in \{1, \ldots, n\}\), \(d_i = (x^{[i]}_i, l_i)\). A regularized empirical risk minimization program with loss function \(LS\) is defined as

\[ w^* \in \arg\min_{w \in \mathbb{R}^r} \frac{C}{n} \sum_{i=1}^{n} LS\left(l_i, g_w\left(x^{[i]}\right)\right) + \frac{1}{2}\|w\|_2^2, \quad (6.22) \]

where \(g_w(x) = \langle \varphi(x), w \rangle\) for a chosen feature mapping \(\varphi: X' \rightarrow \mathbb{R}^r\) taking points from \(X'\) to some (possibly infinite) \(r\)-dimensional feature space, and a hyperplane normal \(w \in \mathbb{R}^r\). Let \(KN(x, y) = \langle \varphi(x), \varphi(y) \rangle\) be the kernel function induced by the feature mapping \(\varphi\).\(^{10}\) The Representer Theorem \(^{58}\) implies that the minimizer \(w^*\) lies in the span of the functions \(KN(\cdot, x^{[i]}_i)\).

\(^{10}\)Note that this is a Reproducing Kernel Hilbert Space (RKHS).
Therefore, we consider \( G : \mathcal{X}^n \times \mathcal{Y} \to \mathbb{R} \) such that
\[
G_d(y) = g_{w^*}(y) = \sum_{i=1}^{n} a_i l_i \text{KN} \left( y, x^{[i]} \right),
\]
for some \( a_i \in \mathbb{R} \). An upper bound on the sensitivity of this function follows from an argument provided by Hall et al. [42] based on a technique of Bousquet and Elisseeff [16]. In particular, we have
\[
\Delta(G) = \sup_{y \in \mathcal{Y}, w \approx w'} |g_w(y) - g_{w'}(y)| \leq \frac{T'C}{n} \sup_{y \in \mathcal{Y}} \text{KN}(y, y),
\]
where \( w \approx w' \) denotes the minimizers of (6.22) which result from input datasets \( d \approx d' \). If \( \text{KN} \) is \((2H, T')\)-smooth, the error introduced is bounded, with probability at least \( 1 - \beta \), by
\[
\tilde{O} \left( \frac{T'C \sup_{y \in \mathcal{Y}} \text{KN}(y, y)}{n \varepsilon} \right)^{1/4}. \]

Note that this result holds with very mild assumptions, namely for any convex and locally \( T'\)-Lipschitz loss function (e.g., square-loss, log-loss, hinge-loss) and any bounded kernel \( \text{KN} \).

Figure 6.2 depicts SVM learning with RBF kernel \((C = \sigma = 1)\) on 1500 each of positive (resp. negative) Gaussian data with mean \([0.3, 0.5]\) (resp. \([0.6, 0.4]\)) and covariance matrix \([0.01, 0; 0, 0.01]\) (resp. \(0.01 \cdot [1, 0.8; 0.8, 1.5] \)), and demonstrates the mechanism’s uniform approximation of predictions, best seen geometrically with the classifier’s decision boundary.

**Logistic Regression.** Let now \( \mathcal{X}' = \{ x \in [0, 1]^\ell : \| x \|_2 \leq 1 \} \). Let furthermore \( \mathcal{X} = \mathcal{X}' \times [0, 1] \) and \( \mathcal{Y} = [0, 1]^{\ell} \). The logistic regressor can be seen as a function \( G : \mathcal{X}^n \times \mathcal{Y} \to \mathbb{R} \) such that \( G_d(y) = \langle w^*, y \rangle \), where \( w^* \) is the minimizer of (6.22) when \( \varphi \) is the identity mapping and the loss function is
\[
\text{LS}(l, \langle w, d \rangle) = \log \left( 1 + e^{-l \langle w, x \rangle} \right).
\]
It is then possible to show that the error introduced by the Bernstein mechanism is bounded, with probability at least \( 1 - \beta \), by
\[
\tilde{O} \left( \frac{C}{n \varepsilon} \right),
\]
since \( G_d(y) \) is a linear function. The prediction with the sigmoid function achieves the same error bound, being a 1/4-Lipschitz function.
6.7 Related Work

Polynomial approximation has proven useful in differential privacy outside function release [80, 18]. Few previous attempts have been made towards private function release. Hall et al. [42] add Gaussian process noise which only yields $(\varepsilon, \delta)$-differential privacy, and does not admit general utility rates. Zhang et al. [89] introduce a functional mechanism for the more specific task of perturbing the objective in private optimization, but they assume separability in the training data and do not obtain rates on utility.

Wang et al. [85] propose a mechanism that releases a summary of data in a trigonometric basis, able to respond to queries that are smooth as in our setting, but are also required to be separable in the training dataset as assumed by Zhang et al. [89]. A natural application is kernel density estimation, which would achieve a rate of $\tilde{O}(1/(n\varepsilon))^{H/(\ell+H)}$ as does our approach. Private KDE has also been explored in various other settings [25] and under weaker notions of utility [42]. Zhang et al. [90] explore discrete naive Bayes under differential privacy, while we investigate parametric Gaussian and non-parametric KDE for class-conditional likelihoods.

As an example of an implicitly defined function, we consider regularized empirical risk minimization such as logistic regression, ridge regression, and the SVM. Previous mechanisms for private SVM release and ERM more generally [19, 75, 21, 48, 47, 8] require finite-dimensional feature mapping or translation-invariant kernels. Hall et al. [42] consider more general mappings but provide $(\varepsilon, \delta)$-differential privacy. Our treatment of regularized ERM extends to kernels that may be translation-variant with infinite-dimensional mappings, while providing stronger privacy guarantees.

6.8 Conclusions and Additional Remarks

In this chapter we considered the release of functions of test data and privacy-sensitive training data. We presented a simple yet effective mechanism for this general setting, that makes use of iterated Bernstein polynomials to approximate any regular function with perturbations applied to the resulting coefficients. Both $\varepsilon$-differential privacy and utility rates are proved in general for the mechanism, with corresponding lower bounds provided. A number of example learners are analyzed, demonstrating the Bernstein mechanism’s versatility.

Similarly to the Laplace and exponential mechanisms, the Bernstein mechanism calibrates the noise addition through the sensitivity of the target non-private function. Unfortunately, bounding sensitivity often involves complex analytic calculation. As an alternative, in a recent paper of the author with Benjamin I. P. Rubinstein [74], a straightforward sampler for estimating sensi-
tivity of non-private mechanisms has been proposed. Since the estimates hold with high probability, combining our sampler with any mechanism that would be differentially private under bounded sensitivity (like the Laplace, exponential, or Bernstein mechanisms) automatically provides random differential privacy—a natural weakening of differential privacy—without any target-specific calculations nor additional mathematical analysis required. Since the relation between differential privacy and other privacy definitions is beyond the scope of this thesis, the technical contributions of the aforementioned paper are not included in this work.
Randomized Response Schemes and the SQ-model

In this chapter, we analyze a special class of mechanisms called randomized response schemes, which permit to provide differential privacy when a database curator cannot be relied upon or trusted. In particular, we introduce the notions of weak and strong usefulness (with respect to a class of queries $\mathcal{F}$) and investigate under which conditions a differentially private randomized response scheme can be strongly useful (resp. cannot even provide weak usefulness). We show that this problem is very much related to the weak SQ-learnability of the class $\mathcal{F}$ under the uniform distribution.

This chapter is based on the paper “Randomized Response Schemes, Privacy and Usefulness” [3], which is a joint work of the author with Hans Ulrich Simon. All results in this chapter were originally published in this paper.

7.1 Introduction

In this chapter, we consider a perspective that is, in some sense, dual to one of learning theory. As shown in Section 2.2 in learning theory we have a sample (a kind of database) and would like to infer some information about an unknown target concept. Here, instead, we assume to have only indirect access to a database via queries. In particular, we restrict our attention to counting queries, which are represented by predicates (like binary concepts in learning theory).

Throughout this chapter, we consider an information consumer who can choose a counting query $f$ (roughly corresponding to a concept) and wishes to “learn” something about an unknown database $d$. Typically, she is not interested in the precise or approximate reconstruction of $d$ but wants to get some piece of aggregated information only (like, for instance, the fraction of
instances in \( d \) satisfying the predicate \( f \) in case of a counting query). With direct access to \( d \), this would be child’s play. However, if \( d \) contains sensitive information, learning something about the data may become more challenging, especially if we aim at providing \( \varepsilon \)-differential privacy. Moreover, the task is even more demanding if we restrict our attention to some very special mechanisms named randomized response schemes \([30]\). They have the great advantage of not relying on a database curator, who may be malicious in general.

This chapter investigates to what extent a randomized response scheme \( M \) can combine \( \varepsilon \)-differential privacy with the conflicting goal of providing useful answers to counting queries taken from a (known) class \( F \). In order to provide \( \varepsilon \)-differential privacy, the database \( d \) is transformed into a “noisy database” \( \tilde{d} = M(d) \). The crucial question is whether one can still extract useful information from \( \tilde{d} \). This general problem is addressed by a series of prior works (see for example \([13, 40]\)), which focus on a worst-case scenario, namely asking that, for all possible choices of the database \( d \), the answers provided by the scheme are close (in probability) to the “correct” ones. It turns out that the family of queries for which this goal can be achieved is quite limited (see Section 7.9 for further details). Here, we relax the requirement on the usefulness (thereby avoiding any trivialization of the problem itself), by analyzing a “minimum” goal (in a sense that will be specified later) which every private randomized response scheme should accomplish. More specifically, we assume that “positive” instances (satisfying the predicate underlying the counting query) entering a database \( d \) are chosen uniformly at random from the set of all positive instances in the universe. A similar remark applies to “negative” instances. An extremely weak notion of “usefulness” is obtained when we merely want to be able to distinguish a database consisting exclusively of negative instances from a database consisting exclusively of positive ones (so that the correct answer to the query would be either 0 or 1). In the seminal work of Dwork et al. \([30]\), it is shown that, for the query class of parities, even this (extremely) weak criterion cannot be satisfied by randomized response schemes that provide \( \varepsilon \)-differential privacy, unless the database becomes exponentially large. In this chapter, we show the following results. First, we extend the just mentioned negative result \([30]\) to any class that is not weakly SQ-learnable under the uniform distribution (which includes the class of parities as a special case). Second, for a broad variety of classes \( F \) that actually are weakly SQ-learnable under the uniform distribution, we design a randomized response scheme that provides \( \varepsilon \)-differential privacy and, at the same time, meets a quite strong criterion of usefulness for every \( f \in F \), namely allowing to infer (in probability) from \( \tilde{d} \) an approximation \( \hat{\omega} \) of the true fraction \( \omega \) of instances in \( d \) satisfying \( f \).

SQ-learnability is a quite influential model with rich relations to other concepts in machine learning theory like, for instance, margin complexity (see Section 2.2.2) and evolvability \([82, 33]\). The results presented in this chapter
(and previous ones [34, 11, 40]) show that these concepts are of high relevance in the field of differential privacy too, so as to establish a strong connection between these two fields.

7.2 Setting

Throughout this chapter, we assume $\mathcal{X}$ to be a finite universe of data records. A counting query over $\mathcal{X}$ is given by a predicate $f : \mathcal{X} \rightarrow \{0, 1\}$.11 Every predicate $f$ splits $\mathcal{X}$ into $\mathcal{X}_{f,0}$ and $\mathcal{X}_{f,1}$ where, for $b = 0, 1$, $\mathcal{X}_{f,b} = \{x \in \mathcal{X} : f(x) = b\}$. The predicate $f$ is called balanced if $|\mathcal{X}_{f,0}| = |\mathcal{X}_{f,1}|$. Throughout this chapter, we also assume that $\mathcal{F}$ is a finite class of counting queries over $\mathcal{X}$. Such a class is called balanced if every $f \in \mathcal{F}$ is balanced. Since both $\mathcal{X}$ and $\mathcal{F}$ are finite, we will often consider the corresponding sign matrix $A = (A_{x,f}) \in \{-1, 1\}^{X \times F}$, where $A_{x,f} = 2f(x) − 1$. Note that this transformation allows us to make use of the notions and results presented in Section 2.2.2.

Given a counting query $f$ and a database $d = (d_1, \ldots, d_n) \in \mathcal{X}^n$, we think of $\bar{f}(d) := \frac{1}{n} \sum_{i=1}^{n} f(d_i)$ as the correct answer.

In this chapter, we consider a particular family of privacy mechanisms, usually referred to as database access mechanisms, which are used to privately generate “noisy answers” to the queries of a given class $\mathcal{F}$. On one hand, the amount of noise should be large enough for providing $\varepsilon$-differential privacy. On the other hand, the noise should not torpedo the usefulness of the answers. In particular, we focus on non-interactive database access mechanisms, meaning that the same mechanism is used for a whole class $\mathcal{F}$ of counting queries. Useful non-interactive mechanisms are harder to design than their interactive counterparts12 but have the advantage that the information $\mathcal{M}(d)$ can be computed once for the whole class $\mathcal{F}$, and then made public. However, for obvious reasons, the party that transforms $d$ into $\mathcal{M}(d)$ must be trusted. In absence of a trusted party, one may need an even more restrictive access mechanism. In this chapter, we aim at analyzing this challenging scenario. In particular, we investigate the so-called randomized response schemes.

**Definition 7.1** (Dwork et al. [30]). A randomized response scheme is a mechanism $\mathcal{M} : \mathcal{X}^n \rightarrow \mathcal{R}^n$ of the form

$$\mathcal{M}(d_1, \ldots, d_n) = (R(d_1), \ldots, R(d_n))$$

for some random map $R : \mathcal{X} \rightarrow \mathcal{R}$.

---

11In contrast to Section 2.2, a counting query is commonly assumed to have range $\{0, 1\}$.
12Where the chosen random map may depend on the actual query, or even on all queries that happened in the past.
Chapter 7. Randomized Response Schemes and the SQ-model

Information consumer

\[ R(d_1) \quad \cdots \quad R(d_{n-1}) \quad R(d_n) \]

Figure 7.1: Randomized response scheme: Each datum \( d_i \) is uploaded in the noisy form \( R(d_i) \) without the need for a trusted party.

Note that, in this case, \( \mathcal{M} \) provides \( \varepsilon \)-differential privacy if and only if \( R \) does (so that the random map \( R \) is forced to blur almost all information in the individual datum \( d_i \)). In real-world applications, a randomized response scheme enables users to upload their record \( d_i \) for analysis in the noisy form \( R(d_i) \) without the need for a trusted party. Figure 7.1 provides a visual representation of a randomized response scheme.

7.3 Outline of the Main Results

Similarly to Section 2.2, we consider the classes of counting queries as indexed by a complexity parameter \( \ell \), i.e., \( \mathcal{F} = \mathcal{F}_\ell \) (resp. \( \mathcal{X} = \mathcal{X}_\ell \)) is a member of a family \( (\mathcal{F}_\ell)_{\ell \geq 1} \) (resp. \( (\mathcal{X}_\ell)_{\ell \geq 1} \)). As for Boolean predicates, \( \ell \) is typically the number of Boolean variables. Furthermore, we always assume that \( \log |\mathcal{F}_\ell| \) and \( \log |\mathcal{X}_\ell| \) are polynomially bounded in \( \ell \). The sign matrix corresponding to \( \mathcal{F}_\ell \) is denoted \( A_\ell \).

Given \( f \in \bigcup_{\ell \geq 1} \mathcal{F}_\ell \), \( 0 \leq \omega \leq 1 \), and \( n \geq 1 \), we define an \((f, \omega, n)\)-random database, denoted by \( X^n_{\omega, f} \), as the outcome of the following random experiment: draw \( \omega n \) instances independently and uniformly at random from \( X_{f,1} \), and draw \( (1 - \omega) n \) instances independently and uniformly at random from \( X_{f,0} \).

Informally, we consider a database access mechanism \( \mathcal{M} \) useful for \( \bigcup_{\ell \geq 1} \mathcal{F}_\ell \) if, for every counting query \( f \), there exists a function \( Q_f \) such that (at least for random databases \( d \)) the “noisy answer” \( Q_f(\mathcal{M}(d)) \) is (in probability) a good approximation of the “correct answer” \( \bar{f}(d) \). Note that \( \bar{f}(d) = \omega \) if \( d \) is an \((f, \omega, n)\)-random database. Moreover, the required database size should be polynomially bounded in \( \ell \) (and in some other relevant variables). This is captured more formally in the following definition.
Definition 7.2 (Strong usefulness). We say that a database access mechanism \( \mathcal{M} \) is strongly useful for the family \( (\mathcal{F}_\ell)_{\ell \geq 1} \) if there exists a function \( \bar{n}(\ell, \alpha, \beta) \) that is polynomially bounded in \( \ell, 1/\alpha, 1/\beta \) and if there exists a family \( (Q_f)_{f \in \mathcal{F}_\ell} \) of (possibly random) maps such that the following holds. For every \( \ell \geq 1 \), for every counting query \( f \in \mathcal{F}_\ell \), for all \( 0 \leq \omega \leq 1 \) and all \( 0 < \alpha, \beta \leq 1/2 \), if \( n \geq \bar{n}(\ell, \alpha, \beta) \), \( d = X^n_{f,\omega} \) and \( \hat{\omega} = Q_f(\mathcal{M}(d)) \), then the probability for \( |\hat{\omega} - \omega| \geq \alpha \) is bounded by \( \beta \). Here the probability is taken over the internal randomization in \( d, \mathcal{M} \) and \( Q_f \).

We refer to \( \bar{n}(\ell, \alpha, \beta) \) as the sample size required by \( \mathcal{M} \). We briefly note that the main difference between our notion of strong usefulness and the original notion of \((\alpha, \beta)\)-usefulness provided by Blum et al. [14] is that we deal with an \( (f, \omega, n) \)-random database whereas their criterion in terms of \( \alpha, \beta \) has to hold for all choices of \( d \).

The definition of weak usefulness is similar except that \( \omega \) is set to either 0 or 1 and it suffices that, for each choice of \( \omega, \beta, \varepsilon, f \), the probability for \( |\hat{\omega} - \omega| < 1/2 \) is at least \( 1 - \beta \) (so that the cases \( \omega = 0 \) and \( \omega = 1 \) can be distinguished with high probability of success).

Definition 7.3 (Weak usefulness). We say that a database access mechanism \( \mathcal{M} \) is weakly useful for the family \( (\mathcal{F}_\ell)_{\ell \geq 1} \) if there exists a function \( \bar{n}(\ell, \beta) \) that is polynomially bounded in \( \ell, 1/\beta \) and if there exists a family \( (Q_f)_{f \in \mathcal{F}_\ell} \) of (possibly random) maps such that the following holds. For every \( \ell \geq 1 \), for every counting query \( f \in \mathcal{F}_\ell \), for all \( \omega \in \{0, 1\} \) and all \( 0 < \beta \leq 1/2 \), if \( n \geq \bar{n}(\ell, \beta) \), \( d = X^n_{f,\omega} \) and \( \hat{\omega} = Q_f(\mathcal{M}(d)) \), then the probability for \( |\hat{\omega} - \omega| \geq 1/2 \) is bounded by \( \beta \).

Dwork et al. [30] show that, for every \( \varepsilon > 0 \), there can be no randomized response scheme for the class of parity functions that provides \( \varepsilon \)-differential privacy and is weakly useful. The main results of this chapter are as follows.

Theorem 7.1. For every \( \varepsilon > 0 \), the following holds. If the margin complexity \( \bar{\gamma}_{\min}(A_\ell)^{-1} \) (cf. Section 2.2.3) associated with a family \( (\mathcal{F}_\ell)_{\ell \geq 1} \) of concept classes is a super-polynomial function in \( \ell \), then \( (\mathcal{F}_\ell)_{\ell \geq 1} \) has no randomized response scheme that provides \( \varepsilon \)-differential privacy and is weakly useful.

The proof of this result is given in Sections 7.4 and 7.5. The proof of the next result is provided in Section 7.6 and makes use of the tools introduced in Section 2.2.2.

Theorem 7.2. For every \( \varepsilon > 0 \), there is a randomized response scheme \( \mathcal{M}_\varepsilon \) for \( (\mathcal{F}_\ell)_{\ell \geq 1} \) that provides \( \varepsilon \)-differential privacy and is strongly useful if at least one of the following conditions is valid:

(i) The classes \( \mathcal{F}_\ell \) are balanced and \( \bar{\gamma}_{\min}(A_\ell)^{-1} \) is polynomially bounded in \( \ell \).
(ii) $\gamma_{\text{min}}^{\text{eff}}(A)\ell$ is polynomially bounded in $\ell$.

Moreover, there is a single polynomial in $\ell, 1/\alpha, 1/\beta, 1/\varepsilon$ (as opposed to a family $(P_\varepsilon)_{\varepsilon>0}$ of polynomials in $\ell, 1/\alpha, 1/\beta$) that bounds from above the sample size required by $\mathcal{M}_\varepsilon$.

It can be shown that Theorem 7.2 applies, for instance, to the class of Boolean monomials, Boolean clauses or, more generally, to Boolean decision lists with a bounded number of label-changes. It furthermore applies to axis-parallel hyper-rectangles over a discrete domain. This is explained in more detail in Section 7.7. Our definitions of weak and strong usefulness ignore efficiency issues. However, we will briefly indicate in Section 7.8 under which conditions the randomized response scheme from Theorem 7.2 can be implemented efficiently. This will specifically be the case for the aforementioned classes.

### 7.4 Proof of Theorem 7.1 for Balanced Classes

In this section, we prove Theorem 7.1 under the additional assumption that the classes $\mathcal{F}_\ell$ are balanced. Our proof will be obtained in a (more or less) straightforward manner from Theorem 7.3 below (being valid for all balanced classes $\mathcal{F}$), which generalizes a similar theorem due to Dwork et al. [30] (being concerned with the class of parity functions only). It turns out that the proof of Theorem 7.3 is quite similar to the proof of the less general result of Dwork et al. [30]. The main new technical contribution is located in the proof of Lemma 7.4 where we apply some algebraic manipulations that bring the Forster bound (cf. Section 2.2.2) into play, which finally leads us to the more general result.

Let $\mathcal{F} = \{f_1, \ldots, f_N\}$ be a class of counting queries over the universe $\mathcal{X} = \{x_1, \ldots, x_M\}$. $A \in \{-1, 1\}^{\mathcal{X} \times \mathcal{F}}$ denotes the corresponding sign matrix. In the sequel, we assume that $\mathcal{F}$ is balanced. The general case is discussed in Section 7.5. Let $q = (q_f)_{f \in \mathcal{F}}$ denote a probability vector. Drawing $f$ at random from $\mathcal{F}$ according to $q$ is denoted by $f \sim \mathcal{F}$. Let $X$ denote the random variable that is uniformly distributed on $\mathcal{X}$. For every $f \in \mathcal{F}$ and every $b = 0, 1$, $X_{f,b}$ denotes the random variable that is uniformly distributed on $\mathcal{X}_{f,b}$. With each query function $f \in \mathcal{F}$ we associate the two quite diverse databases $X_{f,0}^n, X_{f,1}^n \in \mathcal{X}^n$, which (as explained in Section 7.3) are the $(f, \omega, n)$-random databases for $\omega = 0, 1$. An useful answer to a query instantiated by $f$ should be close to $b$ if the underlying (random) database is $X_{f,b}^n$. Suppose that the answer to the query is derived from an $\varepsilon$-differentially private randomized response scheme $\mathcal{M}$ for some $\varepsilon > 0$. The following result indicates that the usefulness of the answer is, in this case, severely limited (at least for query classes whose Forster bound is large).
Theorem 7.3. If $\mathcal{M}$ is an $\varepsilon$-differentially private randomized response scheme for the balanced class $\mathcal{F}$, then (given the above notations) the following holds with probability at least $7/8$ (taken over $f \sim \mathcal{F}^q$):

$$\text{SD} \left( \mathcal{M}(X^n_{f,0}), \mathcal{M}(X^n_{f,1}) \right) \leq 4nC^{1/3} \quad \text{for} \quad C = \frac{(e^\varepsilon - 1)^2}{FB_q(A)^2}. \quad (7.2)$$

Proof. Let $R : \mathcal{X} \rightarrow \mathcal{R}$ be the random map such that $(7.1)$ holds. The following observations can be found in the work of Dwork et al. [30]:

- The $\varepsilon$-differential privacy of $\mathcal{M}$ implies that for all $x, x' \in \mathcal{X}$ and $z \in \mathcal{R}$:
  $$\text{Pr}[R(x) = z] \leq e^\varepsilon \text{Pr}[R(x') = z]. \quad (7.3)$$

- For each $f \in \mathcal{F}$, the following holds. First
  $$\text{SD} \left( \mathcal{M}(X^n_{f,0}), \mathcal{M}(X^n_{f,1}) \right) \leq n \cdot \text{SD} \left( R(X_{f,0}), R(X_{f,1}) \right).$$

  Second
  $$\text{SD} \left( R(X_{f,0}), R(X) \right) = \text{SD} \left( R(X_{f,1}), R(X) \right) = \frac{1}{2} \cdot \text{SD} \left( R(X_{f,0}), R(X_{f,1}) \right).$$

It therefore suffices to show that the probability (taken over $f \sim \mathcal{F}^q$) for

$$\text{SD} \left( R(X_{f,1}), R(X) \right) \leq \left( \frac{C}{\alpha} \right)^{1/3} \quad (7.4)$$

is at least $1 - \alpha$ for every choice of $0 < \alpha < 1$. This sufficient condition actually holds as will become evident from Lemmas 7.4 and 7.5 below. Setting $\alpha = 1/8$, the theorem follows.

The proof of Theorem 7.3 made use of two lemmas that we present now. For the sake of brevity, we define

$$\pi(z|x) := \text{Pr}[R(x) = z], \quad \pi_{f,b}(z) := \text{Pr}[R(X_{f,b}) = z], \quad \pi(z) := \text{Pr}[R(X) = z]. \quad (7.5)$$

Let $B \sim_R \{0, 1\}$ denote the Bernoulli distribution with parameter $1/2$ such that $b \sim B$ is a perfect random bit.

Lemma 7.4. Suppose that $R$ is a random map satisfying condition $(7.3)$. Then, for all $z$ in the range of $R$ and for $C$ as specified in $(7.2)$, the following
Chapter 7. Randomized Response Schemes and the SQ-model

holds:

\[ \mathbb{E}_{f \sim \mathcal{F}, b \sim B}[\pi_{f,b}(z)] = \pi(z) \quad \text{and} \]
\[ \text{Var}_{f \sim \mathcal{F}, b \sim B}[\pi_{f,b}(z)] \leq C \cdot \pi(z)^2. \]  

(7.6)

Proof. Let \( \pi_{\text{min}}(z) = \min_{x \in \mathcal{X}} \pi(z|x) \), \( \bar{\pi}(z|x) = \pi(z|x) - \pi_{\text{min}}(z) \), \( \bar{\pi}_{f,b}(z) = \pi_{f,b}(z) - \pi_{\text{min}}(z) \), and \( \bar{\pi}(z) = \pi(z) - \pi_{\text{min}}(z) \). Let furthermore \( \bar{\pi}(z) \) be the vector \((\bar{\pi}(z|x))_{x \in \mathcal{X}}\). The following observations, partially made by Dwork et al. \cite{30} already, hold for every \( z \) in the range of \( R \):

\[ \forall f \in \mathcal{F} : \mathbb{E}_{b \sim B}[\pi_{f,b}(z)] = \frac{1}{2}(\pi_{f,0}(z) + \pi_{f,1}(z)) = \pi(z) \]

(7.7)

\[ \mathbb{E}_{f \sim \mathcal{F}, b \sim B}[\pi_{f,b}(z)] = \pi(z) \]

(7.8)

\[ \text{Var}_{f \sim \mathcal{F}, b \sim B}[\pi_{f,b}(z)] = \text{Var}_{f \sim \mathcal{F}, b \sim B}[\bar{\pi}_{f,b}(z)] \]

(7.9)

\[ \pi(z|x) \leq e^\varepsilon \pi_{\text{min}}(z) \]

(7.10)

\[ \|\bar{\pi}(z)\|_2^2 \leq M(e^\varepsilon - 1)^2 \pi(z)^2 \]

(7.11)

Equation (7.7) holds because the functions \( f \in \mathcal{F} \) are assumed as balanced. Equation (7.8) is an immediate consequence of (7.7). (7.9) is obvious. (7.10) holds because \( R \) satisfies (7.3). Finally, Inequality (7.11) follows from (7.10) and the following calculation:

\[ \|\bar{\pi}(z)\|_2^2 = \sum_{x \in \mathcal{X}} \bar{\pi}(z|x)^2 = \sum_{x \in \mathcal{X}} (\pi(z|x) - \pi_{\text{min}}(z))^2 \]

\[ \overset{(7.10)}{\leq} M(e^\varepsilon - 1)^2 \pi_{\text{min}}(z)^2 \leq M(e^\varepsilon - 1)^2 \pi(z)^2. \]

For the sake of brevity, we set \( A_q = A \cdot \text{diag}(q)^{1/2} \) and proceed as follows:

\[ \text{Var}_{f \sim \mathcal{F}, b \sim B}[\pi_{f,b}(z)] \overset{(7.9)}{=} \text{Var}_{f \sim \mathcal{F}, b \sim B}[\bar{\pi}_{f,b}(z)] \]

\[ \overset{(7.10)}{=} \mathbb{E}_{f \sim \mathcal{F}, b \sim B}[(\bar{\pi}_{f,b}(z) - \bar{\pi}(z))^2] \]

\[ = \mathbb{E}_{f \sim \mathcal{F}, b \sim B} \left[ \frac{2}{M} \sum_{x \in \mathcal{X}} \bar{\pi}(z|x)\mathbb{1}_{X_{f,b}(x)} - \frac{1}{M} \sum_{x \in \mathcal{X}} \bar{\pi}(z|x) \right]^2 \]

\[ = \mathbb{E}_{f \sim \mathcal{F}, b \sim B} \left[ \left( \frac{1}{M} \sum_{x \in \mathcal{X}} \bar{\pi}(z|x)(2 \cdot \mathbb{1}_{X_{f,b}(x)} - 1) \right)^2 \right] \]

\[ = \mathbb{E}_{f \sim \mathcal{F}, b \sim B} \left[ \left( \frac{1}{M} \sum_{x \in \mathcal{X}} \bar{\pi}(z|x)A_{x,f} \right)^2 \right] \]

\[ = \frac{1}{M^2} \sum_{f \in \mathcal{F}} q_f \left( \sum_{x \in \mathcal{X}} \bar{\pi}(z|x)A_{x,f} \right)^2 \]
7.4. Proof of Theorem 7.1 for Balanced Classes

\[ \begin{align*}
&= \frac{1}{M^2} \sum_{x,x' \in X} \bar{\pi}(z|x) \bar{\pi}(z|x') \sum_{f \in F} q_f A_{x,f} A_{x',f} \\
&= \frac{1}{M^2} \bar{\pi}(z)^\top A_q A_q^\top \bar{\pi}(z) \\
&\leq \frac{1}{M^2} \|\bar{\pi}(z)\|^2_2 \|A_q\|^2_2 - 7.11, 2.2 \leq \left( \frac{(e\varepsilon - 1)\pi(z)}{\text{FB}_q(A)} \right)^2.
\end{align*} \]

The third equation in this calculation results from expanding the definitions of \( \bar{\pi}_{f,b}(z) \) and \( \bar{\pi}(z) \) and from making use of the fact that \( f \) is balanced. The fifth equation holds because \( 2 \cdot 1_{X_{f,b}}(x) - 1 = (2b - 1) \cdot A_{x,f} \). Since the expression containing the factor \( 2b - 1 = \{-1, 1\} \) is squared, the parameter \( b \) vanishes at this stage of the computation. The remaining steps in the calculation are rather obvious.

The following result (resp. a slightly less general version of it) is due to Dwork et al. [30]. For the sake of completeness, we provide the proof below.

**Lemma 7.5.** Let \( \pi(z|x), \pi_{f,b}(z|x) \) and \( \pi(z) \) be given by (7.5), and let \( \mathcal{R} : \mathcal{X} \rightarrow \mathcal{R} \) be a random map such that (1.6) holds for every \( z \) in the range of \( \mathcal{R} \). Let \( 0 < \alpha < 1 \). Then the probability for (7.4), taken over \( f \sim \mathcal{F}_q \), is at least \( 1 - \alpha \).

**Proof.** We say that \( z \) is “\( \xi \)-bad” for \( (f, b) \) if \( |\pi_{f,b}(z) - \pi(z)| > \xi \pi(z) \). Note that \( z \) is \( \xi \)-bad for \( (f, 1) \) if and only if \( z \) is \( \xi \)-bad for \( (f, 0) \) because of (7.7). By Chebyshev’s inequality (see Theorem 2.4),

\[ \Pr_{f \sim \mathcal{F}_q}[z \text{ is } \xi \text{-bad for } (f, 1)] = \Pr_{f \sim \mathcal{F}_q, b \sim \mathcal{B}}[z \text{ is } \xi \text{-bad for } (f, b)] \leq \frac{\text{Var}_{f \sim \mathcal{F}_q, b \sim \mathcal{B}}[\pi_{f,b}(z)]}{\xi^2 \pi(z)^2} \leq \frac{C}{\xi^2} \]

holds for every \( z \). Therefore,

\[ \Pr_{f \sim \mathcal{F}_q, z \sim \mathcal{R}(X)}[z \text{ is } \xi \text{-bad for } (f, 1)] \leq \frac{C}{\xi^2} \quad \text{.} \tag{7.12} \]

Let \( 0 \leq \beta \leq 1 \) be maximal such that there exists \( \mathcal{F}' \subseteq \mathcal{F} \) with the following properties:

- \( \sum_{f \in \mathcal{F}'} q_f \geq \alpha \);
- For every \( f \in \mathcal{F}' \), \( \Pr_{z \sim \mathcal{R}(X)}[z \text{ is } \xi \text{-bad for } (f, 1)] > \beta \).

The definition of \( \beta \) implies that

\[ \Pr_{f \sim \mathcal{F}_q, z \sim \mathcal{R}(X)}[z \text{ is } \xi \text{-bad for } (f, 1)] > \alpha \beta \quad \text{.} \]
In view of (7.12), we may conclude that $\beta < C/(\alpha \xi^2)$. Set now $\beta_0 = C/(\alpha \xi^2)$. It follows that the probability, taken over $f \sim \mathcal{F}_q$, for

$$\Pr_{z \sim R(X)} [z \text{ is not } \xi\text{-bad for } (f, 1)] \geq 1 - \beta_0 \quad (7.13)$$

is at least $1 - \alpha$. Consider now a fixed pair $(f, 1)$ for which (7.13) actually holds. Let $z_{\text{bad}}$ range over the values $z$ that are $\xi$-bad for $(f, 1)$, and let $z_{\text{good}}$ range over the remaining values. We are now ready to bound the statistical difference:

$$\text{SD}(R(X_{f,1}), R(X)) = \frac{1}{2} \sum_z |\pi_{f,1}(z) - \pi(z)|$$

$$= \frac{1}{2} \left( \sum_{z_{\text{good}}} |\pi_{f,1}(z) - \pi(z)| + \sum_{z_{\text{bad}}} |\pi_{f,1}(z) - \pi(z)| \right)$$

$$\leq \frac{1}{2} \left( \xi + \frac{C}{\alpha \xi^2} \right)$$

Setting $\xi = (C/\alpha)^{1/3}$ yields

$$\text{SD}(R(X_{f,1}), R(X)) \leq \left( \frac{C}{\alpha} \right)^{1/3},$$

as desired.

We are now ready to prove Theorem 7.1 under the assumption that the classes $\mathcal{F}_\ell$ are balanced. Let $\mathcal{M}$ be a randomized response scheme for $(\mathcal{F}_\ell)_{\ell \geq 1}$ that provides $\varepsilon$-differential privacy for some $\varepsilon > 0$. We will show that assuming $\mathcal{M}$ to be weakly useful and $\bar{\gamma}_{\text{min}}(\mathcal{A}_\ell)^{-1}$ to be super-polynomial in $\ell$ leads to a contradiction.

Suppose first that $\mathcal{M}$ is weakly useful. Let $\beta = 1/3$. Then, for all $\ell$ and all $f \in \mathcal{F}_\ell$, the criterion for weak usefulness implies that the distributions $X_{f,0}$ and $X_{f,1}$ can be distinguished with probability at least $2/3$. Thus, there will be the same small average error rate of at most $1/3$, when we $q$-average over all $f$ from $\mathcal{F}_\ell$ regardless of how the distribution $q$ on $\mathcal{F}_\ell$ is chosen. Suppose second that $\bar{\gamma}_{\text{min}}(\mathcal{A}_\ell)^{-1}$ is a super-polynomial function in $\ell$. Then, according to Lemma 2.7, $\text{FB}(\mathcal{A}_\ell)$ (as defined in (2.3)) is a super-polynomial function in $\ell$, too. Thus, for every $\ell$, there must exist a probability vector $q = q_\ell$ such that $\text{FB}_{q_\ell}(\mathcal{A}_\ell)$ is a super-polynomial function in $\ell$. According to Theorem 7.3 (applied to $\mathcal{F} = \mathcal{F}_\ell$), Inequality (7.2) is valid with probability (taken over $f \sim \mathcal{F}_q$) at least $7/8$. Since the database size $n$ may grow polynomially in $\ell$ only, the statistical difference between $\mathcal{M}(X_{f,0})$ and $\mathcal{M}(X_{f,1})$ will be negligible (as $\ell$ goes to infinity) for a $q$-fraction $7/8$ (or more) of all $f \in \mathcal{F}_\ell$. We apply Lemma 2.1 and conclude that, for the “bad functions” $f \in \mathcal{F}_\ell$, the Bayes error
for distinguishing between $\mathcal{M}(X_{f,0})$ and $\mathcal{M}(X_{f,1})$ will therefore be arbitrarily close to $1/2$ (as $\ell$ goes to infinity). Thus the Bayes error, $\eta$-averaged over all $f \in \mathcal{F}_\ell$ cannot be significantly smaller than $(7/8) \cdot (1/2) + (1/8) \cdot 0 = 7/16$. Since $7/16 > 1/3$, we arrived at a contradiction.

7.5 Proof of Theorem 7.1

In Section 7.4, we presented a proof of Theorem 7.1 under the additional assumption that the classes $\mathcal{F}_\ell$ are balanced. The proof was obtained in a (more or less) straightforward manner from Theorem 7.3 which presents an upper bound on $\text{SD}(\mathcal{M}(X_{f,0}^n), \mathcal{M}(X_{f,1}^n))$. The main point in the proof was that this upper bound is asymptotically smaller than $1/P(\ell)$ for any polynomial $P$, provided that the margin complexity $\bar{\gamma}_{\min}(\mathcal{A}_\ell)^{-1}$ associated with $(\mathcal{F}_\ell)_{\ell \geq 1}$ is super-polynomial in $\ell$. The main reasons why Theorem 7.1 holds even for unbalanced classes are as follows:

- Theorem 7.3 can be generalized to “almost balanced” classes. See Theorem 7.8 below for the formal statement.

- If the margin complexity $\bar{\gamma}_{\min}(\mathcal{A}_\ell)^{-1}$ associated with $(\mathcal{F}_\ell)_{\ell \geq 1}$ is super-polynomial in $\ell$, then there are almost balanced sub-classes of $\mathcal{F}_\ell$ whose margin complexity is still super-polynomial in $\ell$. See Theorem 7.9 below for the formal statement.

We define the imbalance of a function $f : \mathcal{X} \to \{0, 1\}$ as $\|X_{f,1} - X_{f,0}\|/|\mathcal{X}|$. We say that $\mathcal{F}$ has an imbalance of at most $\Lambda$ if, for each $f \in \mathcal{F}$, the imbalance of $f$ is at most $\Lambda$. Let $\mathcal{X}' \supseteq \mathcal{X}$ be an extended universe such that $|\mathcal{X}'| = |\mathcal{X}| + \Lambda|\mathcal{X}|$. Clearly, the functions $f$ of a class with an imbalance of at most $\Lambda$ can be extended to the larger domain $\mathcal{X}'$ so as to become (completely) balanced. This balanced extension of $f$ is denoted $f'$ so that $\mathcal{F}' = \{f' : f \in \mathcal{F}\}$ is a balanced extension of $\mathcal{F}$. With these notations, the following holds.

**Lemma 7.6.** Let $\mathcal{F}$ have imbalance of at most $\Lambda$. Then, for all $f \in \mathcal{F}$, we have:

$$\text{SD}(X_{f',0}, X_{f,0}) + \text{SD}(X_{f',1}, X_{f,1}) \leq 4\Lambda.$$  

**Proof.** We recall that, for $b \in \{0, 1\}$, $X_{f,b}$ (resp. $X_{f',b}$) denotes the random variable that is uniformly distributed on $\mathcal{X}_{f,b}$ (resp. $\mathcal{X}'_{f,b}$). We thus have

$$\text{SD}(X_{f',b}, X_{f,b}) \leq \frac{1}{2} \left( 2 \cdot \frac{\Lambda|\mathcal{X}|}{|\mathcal{X}_b| + ||\mathcal{X}_{f,1} - |\mathcal{X}_{f,0}|/|\mathcal{X}|)} \right) \leq \frac{\Lambda|\mathcal{X}|}{|\mathcal{X}|/2} = 2\Lambda.$$

$\square$
**Lemma 7.7.** Suppose that $\mathcal{F}$ has an imbalance of at most $\Lambda$. Let $\mathcal{F}'$ be its balanced extension, and let $A$ and $A'$ be the corresponding sign matrices, respectively. Let $q$ be a probability vector for the functions in $\mathcal{F}$. With these notations, the following holds:

\[
\frac{1}{\text{FB}_q(A')^2} \leq \frac{1}{\text{FB}_q(A)^2} + \frac{2\sqrt{\Lambda}}{\text{FB}_q(A)} + \Lambda.
\]

**Proof.** For the sake of brevity, we set $A_q = A \cdot \text{diag}(q)^{1/2}$. We may think of $A'$ as having the sign matrix $A$ (with $|X|$ rows) in its upper block and another sign matrix, say $E$, with $\Lambda|X|$ rows in its lower block. We recall that $M = |X|$.

With these notations, the spectral norm of $A'_q$ can be bounded from above as follows:

\[
\|A'_q\|_2 \leq \|A_q\|_2 + \|E_q\|_2 \leq \|A_q\|_2 + \sqrt{\Lambda M}
\]

Thus, we obtain

\[
\frac{1}{\text{FB}_q(A')^2} \leq \frac{\|A'_q\|_2^2}{(1 + \Lambda)M} \leq \frac{(\|A_q\|_2 + \sqrt{\Lambda M})^2}{(1 + \Lambda)M} \leq \frac{\|A_q\|_2^2}{M} + \frac{2\Lambda M \|A_q\|_2}{M} + \frac{\Lambda M}{M} \leq \frac{1}{\text{FB}_q(A)^2} + \frac{2\sqrt{\Lambda}}{\text{FB}_q(A)} + \Lambda,
\]

which concludes the proof. \qed

We can now introduce the following extension of Theorem 7.3.

**Theorem 7.8.** Suppose that $\mathcal{F}$ has an imbalance of at most $\Lambda$. Let $\mathcal{F}'$ be its balanced extension, and let $A$ and $A'$ be the corresponding sign matrices, respectively. Let $q$ be a probability vector for the functions in $\mathcal{F}$, and let $\varepsilon > 0$. If $M$ is an $\varepsilon$-differentially private randomized response scheme, then the following holds with probability at least $7/8$ (taken over $f \sim \mathcal{F}'$):

\[
\text{SD}(M(X^n_{f,0}), M(X^n_{f,1})) \leq 4\Lambda n + 4n(\varepsilon^2 - 1)^2 \cdot C',
\]

where

\[
C' = \frac{1}{\text{FB}_q(A)^2} + \frac{2\sqrt{\Lambda}}{\text{FB}_q(A)} + \Lambda.
\]
7.5. Proof of Theorem 7.1

Proof. According to the triangle inequality, $\text{SD}(\mathcal{M}(X^n_{f,0}), \mathcal{M}(X^n_{f,1}))$ is upper-bounded by

$$
\sum_{b=0,1} \text{SD}(\mathcal{M}(X^n_{f,b}), \mathcal{M}(X^n_{f',b})) + \text{SD}(\mathcal{M}(X^n_{f,0}), \mathcal{M}(X^n_{f,1}))
$$

The first sum is upper-bounded by

$$
\sum_{b=0,1} \text{SD}(X^n_{f,b}, X^n_{f',b}) \leq n \sum_{b=0,1} \text{SD}(X_{f,b}, X_{f',b})
$$

and therefore, by virtue of Lemma 7.6 by $4\Lambda n$. According to Theorem 7.3, the following holds with probability at least $7/8$ (taken over $f \sim F^q$):

$$
\text{SD}(\mathcal{M}(X^n_{f,0}), \mathcal{M}(X^n_{f',1})) \leq 4n(e^{\epsilon} - 1)^2 \frac{1}{\text{FB}_q(A')^2}.
$$

Plugging in the upper bound on $1/\text{FB}_q(A')^2$ from Lemma 7.7 and putting all pieces together, the theorem follows.

Here comes the final piece of the puzzle in our proof of Theorem 7.1.

**Theorem 7.9.** If the margin complexity $\bar{\gamma}_{\text{min}}(A_{\ell})^{-1}$ associated with a family $(\mathcal{F}_\ell)_{\ell \geq 1}$ of concept classes is a super-polynomial function in $\ell$, then, for each polynomial $P(\ell) \geq 1$, there exists a family of sub-classes $\mathcal{F}_\ell^P \subseteq \mathcal{F}_\ell$ such that the following holds:

(i) For each $\ell \geq 1$, the imbalance of $\mathcal{F}_\ell^P$ is bounded by $1/P(\ell)$.

(ii) The margin complexity $\bar{\gamma}_{\text{min}}(A^P_{\ell})^{-1}$ associated with the family $(\mathcal{F}_\ell^P)_{\ell \geq 1}$ is a super-polynomial function in $\ell$.

Proof. We simply choose $\mathcal{F}_\ell^P$ as the set of all $f \in \mathcal{F}_\ell$ whose imbalance is bounded by $1/P(\ell)$. Let $\ell$ be arbitrary but fixed. Let $\mathcal{F} = \mathcal{F}_\ell$ and $A = A_{\ell}$. Each, say $r$-dimensional, arrangement $A$ for the sign matrix $A$ can be extended as follows:

- Add an extra dimension $r + 1$.
- If $f \in \mathcal{F}^P$, then the coordinate $r + 1$ of the vector $v[f]$ is set to 0.
- If $f \in \mathcal{F} \setminus \mathcal{F}^P$ is biased towards positive (resp. negative) examples, then the coordinate $r + 1$ of $v[f]$ gets value 1 (resp. $-1$), and the remaining coordinates are set to 0.
- The coordinate $r + 1$ of each vector $u[x]$ is set to 1.
• For each \( x \in \mathcal{X} \), the vector \( u^x \) is normalized so as to have unit Euclidean norm (by applying the scaling factor \( 1/\sqrt{2} \)).

The effect is as follows:

• For each function \( f \in F \setminus F^P \), the average margin achieved by the extension of \( A \) is at least \( (\sqrt{2}P(\ell))^{-1} \).

• For each function \( f \in F^P \), the average margin achieved by the extension of \( A \) coincides with \( 1/\sqrt{2} \) times the average margin achieved by the original arrangement \( A \).

Since \( \bar{\gamma}_{\min}(A, \ell) - 1 \) is super-polynomial in \( \ell \) but the functions from \( F \setminus F^P \) cannot be blamed for it, it follows that \( \bar{\gamma}_{\min}(A, \ell) - 1 \) is super-polynomial in \( \ell \).

The proof of Theorem 7.8 can now be completed in a similar fashion as it was done at the end of Section 7.4 for the special case of balanced classes.

### 7.6 Proof of Theorem 7.2

The analysis of the randomized response scheme that we are going to design requires Hoeffding’s inequality (see Theorem 2.5), and, in addition, the following tail bound.

**Theorem 7.10.** For \( i = 1, \ldots, n \) and \( j = 1, \ldots, r \), let \( Z_{i,j} \sim \text{Lap}(\lambda) \) be i.i.d. random variables. For each \( i = 1, \ldots, n \), let \( v^i = (v^i_j)_{j=1,\ldots,r} \) be a vector of unit Euclidean norm, i.e., \( \|v^i\|_2 = 1 \). Then, for each \( 0 < \xi \leq \sqrt{8} \lambda n \), the following holds:

\[
\Pr \left[ \sum_{i=1}^{n} \sum_{j=1}^{r} v^i_j Z_{i,j} \geq \xi \right] \leq \exp \left( -\frac{\xi^2}{8n \lambda^2} \right).
\]

**Proof.** Let \( S = \sum_{i=1}^{n} \sum_{j=1}^{r} v^i_j Z_{i,j} \). By Markov’s inequality (see Theorem 2.3), we get

\[
\Pr[S > \xi] = \Pr[ e^{tS} > e^{t\xi}] \leq \frac{\mathbb{E}[e^{tS}]}{e^{t\xi}}
\]

for every \( t > 0 \). Note that \( m_S(t) = \mathbb{E}[e^{tS}] \) is the moment generating function of \( S \) and \( m_Z(t) = \mathbb{E}[e^{tZ}] = 1/(1 - (\lambda t)^2) \), defined for \( t < 1/\lambda \), is the moment generating function of \( Z \sim \text{Lap}(\lambda) \) \[59\]. Suppose that \( (\lambda t)^2 \leq 1/2 \) so that \( 1 - (\lambda t)^2 > \exp(-2(\lambda t)^2) \). Making use of the assumption \( \|v^i\|_2 = 1 \), it follows that

\[
m_S(t) = \prod_{i=1}^{n} \prod_{j=1}^{r} \frac{1}{1 - (v^i_j \lambda t)^2}.
\]
7.6. Proof of Theorem 7.2

\[
< \prod_{i=1}^{n} \prod_{j=1}^{r} \frac{1}{\exp \left( -2 \left( v_{i,j}^i \lambda t \right)^2 \right)} = \exp \left( 2(\lambda t)^2 n \right).
\]

Setting \( t = \xi/(4\lambda^2 n) \), we conclude that

\[
\Pr[S > \xi] \leq \frac{mS(t)}{e^{\xi}} < \exp \left( 2\lambda^2 t^2 n - t \xi \right) = \exp \left( -\frac{\xi^2}{8n\lambda^2} \right),
\]

as desired. Note that the assumption \( \xi \leq \sqrt{8}\lambda n \) makes sure that the constraint \((\lambda t)^2 \leq 1/2\), a constraint that we needed in the course of our proof, actually holds for our final choice \( t = \xi/(4\lambda^2 n) \).

Setting \( v[i] = (1, 0, \ldots, 0) \) for \( i = 1, \ldots, n \), Theorem 7.10 collapses to the following result.

**Corollary 7.11.** For \( i = 1, \ldots, n \), \( Z_i \sim \text{Lap}(\lambda) \) be i.i.d. random variables. Then, for each \( 0 < \xi \leq \sqrt{8}\lambda n \), the following holds:

\[
\Pr \left[ \sum_{i=1}^{n} Z_i \geq \xi \right] \leq \exp \left( -\frac{\xi^2}{8n\lambda^2} \right) \tag{7.14}
\]

We would like to note that a result similar to Corollary 7.11 can be found in the work of Kasiviswanathan et al. [54]. The proof there, however, is slightly flawed, although it can easily be fixed at the expense of a slightly weaker statement. Our proof of Theorem 7.10 actually builds upon the proof given by Kasiviswanathan et al. [54], except for the slightly flawed part.

Note that the assumption \( \xi \leq \sqrt{8}\lambda n \) in Theorem 7.10 is not very restrictive because, for \( \xi \geq \sqrt{8}\lambda n \), we get

\[
\Pr \left[ \sum_{i=1}^{n} \sum_{j=1}^{r} v_{j,i}^i Z_{i,j} \geq \xi \right] \leq \Pr \left[ \sum_{i=1}^{n} \sum_{j=1}^{r} v_{j,i}^i Z_{i,j} \geq \sqrt{8}\lambda n \right] \leq e^{-n}.
\]

We are now ready for the central part of this section. In order to make our notation as clear as possible, we henceforth denote random vectors as bold uppercase letters, e.g., \( Z \). As usual, let \( \mathcal{F} \) denote a class of counting queries over the universe \( \mathcal{X} \). \( \mathbf{A} \in \{-1, 1\}^{\mathcal{X} \times \mathcal{F}} \) denotes the corresponding sign matrix. Let \( \mathcal{A} = \left( (\mathbf{u}[x])_{x \in \mathcal{X}}, (\mathbf{v}[f])_{f \in \mathcal{F}} \right) \) be an \( r \)-dimensional arrangement for \( \mathbf{A} \). We aim at designing a randomized response scheme that is strongly useful and provides \( \varepsilon \)-differential privacy. To this end, we define the random map \( R : \mathcal{X} \rightarrow \mathbb{R}^r \).
The random vectors $Z$ are chosen independently for different choices of $x$.

**Lemma 7.12.** Let $\epsilon > 0$. Choose the random map $R$ according to (7.15) and $M$ according to (7.1). Then, $M$ provides $\epsilon$-differential privacy.

**Proof.** Let $d' \in X^n$ be a second database differing from $d$ in one entry only, say in entry $d'_k \neq d_k$. Consider the map $g(d) = (u^{[d_1]}, \ldots, u^{[d_n]}) \in (\mathbb{R}^r)^n$. Then
\[
\|g(d) - g(d')\|_1 = \|u^{[d_k]} - u^{[d'_k]}\|_1 \leq \sqrt{r} \|u^{[d_k]} - u^{[d'_k]}\|_2 \leq 2\sqrt{r},
\]
since $u^{[d_k]}$ and $u^{[d'_k]}$ have unit Euclidean norm. It follows that the sensitivity of $g$ is bounded by $2\sqrt{r}$. According to Lemma 3.4 (applied with $nr$ in place of $r$), the mechanism $M$ provides $\epsilon$-differential privacy.

For the sake of simplicity, we will use the abbreviations $\tilde{\gamma}(f) = \tilde{\gamma}_f(A|A)$ and $\tilde{\gamma}_{\min} = \tilde{\gamma}_{\min}(A|A)$ in what follows. As in Definition 7.2, $X^n_{f,\omega} \in X^n$ denotes an $(f, \omega, n)$-random database. We will show that a good guess $\hat{\omega}$ for $\omega$ can be derived (in probability) from $M(d)$, where $M$ is the random map from Lemma 7.12. To this end, let $f \in \mathcal{F}$ be a counting query and $v = v[f]$ its representation in the arrangement $A$. The following margin parameters, associated with $f$ (resp. $v$) and $b = 0, 1$, will play a central role:
\[
\tilde{\gamma}_b(f) = \frac{1}{|A_{f,b}|} \sum_{x \in A_{f,b}} (2b - 1) \langle u^{[x]}, v \rangle, \quad (7.16)
\]
\[
\tilde{\gamma}(f) = \frac{1}{2} (\tilde{\gamma}_0(f) + \tilde{\gamma}_1(f)). \quad (7.17)
\]

If the random database $d = X^n_{f,\omega}$ contains the instances $X_1, \ldots, X_n$, then its noisy version $\hat{d} = M(d)$ has the following form:
\[
\hat{d} = \left(\hat{X}_1, \ldots, \hat{X}_n\right) \quad \text{for} \quad \hat{X}_i = R(X_i) = u^{[X_i]} + Z_i
\]
\[
\text{and} \quad Z_i \sim \text{Lap} \left(\frac{2\sqrt{r}}{\epsilon}\right). \quad (7.15)
\]

Consider the random variable $S = S_1 + S_2$ for
\[
S_1 = \frac{1}{n} \sum_{i=1}^{n} \langle u^{[X_i]}, v \rangle \quad \text{and} \quad S_2 = \frac{1}{n} \sum_{i=1}^{n} \langle Z_i, v \rangle. \quad (7.18)
\]
Now, we obtain
\[
\mathbb{E}[S] = \mathbb{E}[S_1] = \omega \bar{\gamma}_1(f) - (1 - \omega) \bar{\gamma}_0(f) \\
= \omega (\bar{\gamma}_0(f) + \bar{\gamma}_1(f)) - \bar{\gamma}_0(f).
\]
Solving for \(\omega\) yields
\[
\omega = \frac{\mathbb{E}[S] + \bar{\gamma}_0(f)}{\bar{\gamma}_0(f) + \bar{\gamma}_1(f)}.
\]

Treating \(S\) as a (hopefully good) approximation of \(\mathbb{E}[S]\), this motivates the following definition of \(\bar{\omega}\):

1. Given \(d\), compute \(S = S_1 + S_2 = \frac{1}{n} \sum_{i=1}^{n} \langle \hat{X}_i, v \rangle\).
2. Compute \(\bar{\omega} = Q_f(d)\) according to the right hand-side of (7.19) except that \(S\) is substituted for (the unknown) \(\mathbb{E}[S]\).

View now \(F = F_\ell\) as a member of the parameterized class \((F_\ell)_{\ell \geq 1}\). Remember that we assume \(\log |X_\ell|\) and \(\log |F_\ell|\) to be polynomially bounded in \(\ell\). Note that \(\tilde{\gamma}(f) = \bar{\gamma}(f)\) if \(f\) is balanced, and \(\tilde{\gamma}(f) \leq 2\bar{\gamma}(f)\) if the arrangement \(A\) is error-free. Therefore, Theorem 7.2 will be an immediate consequence of the following two results.

**Theorem 7.13.** Let \(\varepsilon > 0\). The randomized response scheme, given by \(R\) and \((Q_f)_{f \in F}\) as defined above, provides \(\varepsilon\)-differential privacy. Moreover, it is strongly useful provided that the size \(n = n(r, \alpha, \beta, \varepsilon, \tilde{\gamma}_{\min})\) of the (random) database satisfies

\[
n \geq \frac{32 \cdot r}{\alpha^2 \varepsilon^2 \tilde{\gamma}_{\min}^2} \ln \left( \frac{2}{\beta} \right)
\]

(7.20) where \(\tilde{\gamma}_{\min} = \min_{f \in F} \tilde{\gamma}(f)\).

**Proof.** We first observe that \(\varepsilon\)-differential privacy is granted by Lemma 7.12. We have to show that, with probability at least \(1 - \beta\), \(|\bar{\omega} - \omega| \leq \alpha\) where \(\bar{\omega}\) is computed according to the right hand-side of (7.19) except that \(S\) is substituted for \(\mathbb{E}[S]\). An inspection of (7.19) shows that the condition \(|\bar{\omega} - \omega| \leq \alpha\) is equivalent to \(|S - \mathbb{E}[S]| \leq \alpha (\tilde{\gamma}_0(f) + \tilde{\gamma}_1(f))\). Since \(S = S_1 + S_2\) and \(2\tilde{\gamma}_{\min} \leq 2\bar{\gamma}(f) = \tilde{\gamma}_0(f) + \tilde{\gamma}_1(f)\), the latter condition is guaranteed if the following holds:

\[
|S_1 - \mathbb{E}[S_1]| \leq \alpha \tilde{\gamma}_{\min} \quad \text{and} \quad |S_2 - \mathbb{E}[S_2]| \leq \alpha \tilde{\gamma}_{\min}
\]

(7.21) According to (7.18), \(S_1\) is the average over the independent random variables \(\langle u^{[X_\ell]}, v \rangle \in [-1, 1]\). An application of Hoeffding’s inequality (see Theorem 2.5) shows that the probability for the first condition in (7.21) to be violated is bounded by

\[
2 \exp \left( - \frac{\alpha^2 \tilde{\gamma}_{\min}^2 n}{2} \right)
\]
According to Theorem 7.10, the probability for the second condition in (7.21) to be violated is bounded by
\[
2 \exp \left( -\frac{\alpha^2 \tilde{\gamma}^2 \varepsilon^2 n}{32 \cdot r} \right),
\]
a bound which is more restrictive than the preceding one.\(^{13}\) Setting
\[
2 \exp \left( -\frac{\alpha^2 \tilde{\gamma}^2 \varepsilon^2 n}{32 \cdot r} \right) \leq \beta
\]
and solving for \( n \), we see that the database size \( n \), as specified in (7.20), is sufficiently large.

In Theorem 7.2, it is assumed that there are arrangements \( A_\ell \) for the classes \( F_\ell \) such that the margin complexity \( \bar{\gamma}_{\text{min}}(A_\ell | A_\ell) \) is polynomially bounded in \( \ell \). As already noted above, \( \gamma_{\text{min}} \) and \( \bar{\gamma}_{\text{min}} \) are equal up to a factor of at most 2 under the assumption (made in Theorem 7.2) that the classes \( F_\ell \) are balanced or that the arrangements \( A_\ell \) are error-free. An inspection of (7.20) shows that the only remaining parameter with a possibly super-polynomial growth in \( \ell \) (or in the other relevant parameters) is the dimension \( r = r(F_\ell) \) of the arrangement. However, \( r \) can be assumed to be small because of the following result, whose proof is based on random projection techniques \([50, 6, 9]\).

**Theorem 7.14.** Given a (possibly high-dimensional) arrangement \( A \) for \( F \) such that \( \gamma := \tilde{\gamma}_{\text{min}}(A | A) > 0 \), there exists an \( r \)-dimensional arrangement \( A' \) for \( F \) such that \( \tilde{\gamma}_{\text{min}}(A | A') \geq \gamma/8 \) provided that
\[
r \geq 8 \ln \left( \frac{4(M + N)}{\beta} \right) + \frac{32}{\gamma^2} \ln \left( \frac{8N}{\beta} \right).
\]  
(7.22)

**Proof.** We will make use of the following two results.

**Lemma 7.15 (Arriaga and Vempala \([6]\)).** Let \( u \in \mathbb{R}^s \) be arbitrary but fixed. Let \( T = (T_{i,j}) \) be a random \((r \times s)\)-matrix such that the entries \( T_{i,j} \) are i.i.d. according to the normal distribution \( \mathcal{N}(0,1) \). Consider the random projection \( u' := \frac{1}{\sqrt{r}}(Tu) \in \mathbb{R}^r \). Then the following holds for every constant \( \xi > 0 \):
\[
P_T \left[ \|u'\|^2 \geq ||u||^2 + \xi \|u\|^2 \right] \leq 2 \exp \left( -\frac{\xi^2 r}{8} \right).
\]

\(^{13}\)Note that the application of Theorem 7.10 is justified because the assumption made there, namely \( \xi \leq \sqrt{8} \lambda n \), is granted in our application where \( \xi = \alpha \tilde{\gamma}_{\text{min}} \leq \alpha \leq 1 \) and \( \lambda = 2\sqrt{r}/\varepsilon \geq 2 \).
7.6. Proof of Theorem 7.2

Lemma 7.16 (Ben-David et al. [9]). Let \( \mathbf{v}, \mathbf{x} \in \mathbb{R}^s \) be arbitrary but fixed. Let \( \mathbf{T} \) be the same random \((r \times s)\)-matrix and let \( \mathbf{u} \mapsto \mathbf{u}' \) be the same random projection as in Lemma 7.15. Then the following holds for every constant \( \xi > 0 \):

\[
\mathbb{P}_T \left[ |\langle \mathbf{v}', \mathbf{x}' \rangle - \langle \mathbf{v}, \mathbf{x} \rangle| \geq \frac{\xi}{2} \left( \|\mathbf{v}\|_2^2 + \|\mathbf{x}\|_2^2 \right) \right] \leq 4 \exp \left( -\frac{\xi^2 r}{8} \right).
\]

We are now prepared for the proof of Theorem 7.14. Let \( M := |\mathcal{X}| \) be the number of instances in the universe, and let \( N := |\mathcal{F}| \) be the number of counting queries in \( \mathcal{F} \). Suppose that \( \mathcal{A} \) is an \( s \)-dimensional arrangement for \( \mathcal{F} \) (resp. for the corresponding sign matrix \( \mathbf{A} \)) such that \( \gamma := \tilde{\gamma}_{\text{min}}(\mathbf{A}|\mathcal{A}) > 0 \). Let \( \mathcal{A}' \) be the \( r \)-dimensional arrangement for \( \mathcal{F} \) that results from \( \mathcal{A} \) by randomly projecting every vector \( \mathbf{u} \in \{\mathbf{u}[x] : x \in \mathcal{X}\} \) (resp. every vector \( \mathbf{v} \in \{\mathbf{v}[f] : f \in \mathcal{F}\} \)) to \( \mathbf{u}' \) (resp. to \( \mathbf{v}' \)). Recall that the vectors \( \mathbf{u}[x], \mathbf{v}[f] \) have unit Euclidean norm. Let \( \pi_1 \) denote the probability for

\[
\exists \mathbf{w} \in \{\mathbf{u}[x] : x \in \mathcal{X}\} \cup \{\mathbf{v}[f] : f \in \mathcal{F}\} : \|\mathbf{w}'\|_2 \geq 2.
\] (7.23)

According to Lemma 7.15, applied with 1 in the role of \( \xi \) and combined with the union bound, we get \( \pi_1 \leq 2(M + N) e^{-r/8} \). For every \( f \in \mathcal{F} \), consider the vector

\[
\mathbf{x}[f] = \frac{1}{2} \cdot \sum_{b=0,1} \left( \frac{1}{|\mathcal{X}_{f,b}|} \sum_{x \in \mathcal{X}_{f,b}} A_{x,f} \mathbf{u}[x] \right).
\]

Note that \( \|\mathbf{x}[f]\|_2 \leq 1 \). Let \( \pi_2 \) denote the probability for

\[
\exists f \in \mathcal{F} : \left| \langle \mathbf{v}[f]', \mathbf{x}[f]' \rangle - \langle \mathbf{v}[f], \mathbf{x}[f] \rangle \right| \geq \frac{\gamma}{2}.
\] (7.24)

Note that \( \langle \mathbf{v}[f], \mathbf{x}[f] \rangle = \tilde{\gamma}_f(\mathbf{A}|\mathcal{A}) \geq \gamma \) for all \( f \in \mathcal{F} \). Moreover,

\[
\frac{\gamma}{2} = 2 \cdot \frac{\gamma}{4} \geq \left( \|\mathbf{v}[f]\|_2^2 + \|\mathbf{x}[f]\|_2^2 \right) \cdot \frac{\gamma}{4}.
\]

According to Lemma 7.16 applied with \( \gamma/2 \) in the role of \( \xi \) and combined with the union bound, we get \( \pi_2 \leq 4N e^{-\gamma^2/32} \). As an easy calculation shows, condition (7.22) implies that, with probability at least \( 1 - \beta \), none of the events (7.23), (7.24) occurs. If none of these events occurs, then

\[
\tilde{\gamma}_{\text{min}}(\mathbf{A}|\mathcal{A}') = \min_{f \in \mathcal{F}} \tilde{\gamma}_f(\mathbf{A}|\mathcal{A}') \geq (1/4)(\gamma/2) = \gamma/8,
\]

87
where the scaling factor $1/4$ takes into account that the vectors in the arrangement $\mathcal{A}'$ (with an Euclidean norm of at most 2) must still be normalized so as to have unit Euclidean norm.

Chapter 7. Randomized Response Schemes and the SQ-model

7.7 Arrangements with a Non-Negligible Minimum Margin

In this section, we start with a (more or less trivial) observation, Lemma 7.17 below, which is applied afterward to a couple of Boolean concept classes.

Let $\mathbf{A} \in \{-1,1\}^{M \times N}$ be a sign matrix and let $\mathcal{A}$ be an arrangement for $\mathbf{A}$. The margin parameter $\bar{\gamma}_{\text{min}}(\mathbf{A} | \mathcal{A})$ that is used in Theorems 7.1 and 7.2 is clearly bounded from below by

$$\gamma_{\text{min}}(\mathbf{A} | \mathcal{A}) = \min\{\gamma_{i,j}(\mathbf{A} | \mathcal{A}) : i = 1, \ldots, M \text{ and } j = 1, \ldots, N\} ,$$

which is the minimum among all $MN$ individual margin parameters.

Let $(\mathcal{F}_\ell)_{\ell \geq 1}$ be a family of concept classes, and let $\mathbf{A}_\ell \in \{-1,1\}^{X_\ell \times \mathcal{F}_\ell}$ be the sign matrix associated with $\mathcal{F}_\ell$. We say that $(\mathcal{F}_\ell)_{\ell \geq 1}$ has a non-negligible minimum margin if there exists a polynomial $P(\ell) \in \mathbb{N}$, and, for all $\ell \geq 1$, an arrangement $\mathcal{A}_\ell$ such that $\gamma_{\text{min}}(\mathbf{A}_\ell | \mathcal{A}_\ell) \geq 1/P(\ell)$. This clearly implies that each arrangement $\mathcal{A}_\ell$ for $\mathbf{A}_\ell$ is error-free and that $\gamma_{\text{min}}(\mathbf{A}_\ell | \mathcal{A}_\ell) \leq \bar{\gamma}_{\text{ef}}_{\text{min}}(\mathbf{A}_\ell | \mathcal{A}_\ell)$. It follows that the conclusion of Theorem 7.2 holds, in particular, for all classes with a non-negligible minimum margin.

For each $\ell \geq 1$, let $\mathcal{F}_\ell$ be a class of concepts over the Boolean domain $X_\ell = \{0,1\}^\ell$ or $X_\ell = \{-1,1\}^\ell$. We say that $(\mathcal{F}_\ell)_{\ell \geq 1}$ is linearly separable with polynomially bounded weights if there exists a polynomial $P(\ell) \in \mathbb{N}$, and, for all $\ell \geq 1$ and all $f \in \mathcal{F}_\ell$, a weight vector $w^{[f]} \in \{-P(\ell), \ldots, -1, 0, 1, \ldots, P(\ell)\}^\ell$ together with a threshold $t \in \mathbb{R}$ such that, for all $x \in X_\ell$,

$$f(x) = 1 \iff \langle w^{[f]}, x \rangle \geq t . \quad (7.25)$$

Assume now that this is the case. Clearly, the threshold $t$ can w.l.o.g. be chosen from the interval $[-\ell P(\ell), \ell P(\ell)]$. It follows that $(\mathcal{F}_\ell)_{\ell \geq 1}$ has a non-negligible minimum margin. We could, for example, represent $x$ and $f$ by the
(\ell+1)-dimensional vectors

\[ u^x = \left( \sqrt{P(\ell)} \cdot x, \sqrt{\ell P(\ell)} \right) \quad \text{and} \]
\[ v^f = \left( \frac{1}{\sqrt{P(\ell)}} \cdot w^f, -\frac{t}{\sqrt{\ell P(\ell)}} \right) \]

respectively, so that \( \langle v^f, u^x \rangle = \langle w^f, x \rangle - t \). Moreover, \( t \) can be chosen in the middle of two consecutive non-negative integers. Denote the arrangement obtained by this setting as \( A_\ell \). Note that \( \|v^f\|_2, \|u^x\|_2 \leq \sum 2\ell P(\ell) \).

After normalizing all vectors to unit Euclidean norm, we obtain \( \gamma_{\min}(A_\ell | A_\ell) \geq \frac{1}{4\ell P(\ell)} \). We arrive at the following result.

**Lemma 7.17.** Let \( P(\ell) \) be a polynomial. If \((\mathcal{F}_\ell)_{\ell \geq 1}\) is a family of Boolean concept classes that is linearly separable with integer weights of absolute value at most \( P(\ell) \), then there exist arrangements \((A_\ell)_{\ell \geq 1}\) such that

\[ \gamma_{\min}(A_\ell | A_\ell) \geq \frac{1}{4\ell P(\ell)} . \]

We now discuss some applications. Hereafter, we make use of the following notation. For a Boolean variable \( z \), we define \( z^1 := z \) and \( z^0 := \neg z \), i.e., the negation of \( z \).

**Boolean Monomials.** A monomial over the Boolean variables \( z_1, \ldots, z_\ell \) is given by a conjunction of literals

\[ z_{i(1)}^{\sigma_1} \land z_{i(2)}^{\sigma_2} \land \cdots \land z_{i(s)}^{\sigma_s} , \]

where \( 1 \leq s \leq \ell \), \( i(1), \ldots, i(s) \in \{1, \ldots, \ell\} \) and \( \sigma_1, \ldots, \sigma_s \in \{0, 1\} \). A monomial represents a Boolean function that assigns the bit 1 to a vector \( x \in \mathcal{X}_\ell = \{0, 1\}^\ell \) if and only if \( x_{i(1)}^{\sigma_1} = \cdots = x_{i(s)}^{\sigma_s} = 1 \). A Boolean monomial is linearly separable with weights from \( \{-1, 0, 1\} \) (in the obvious fashion). Thus, Lemma 7.17 applies with \( P(\ell) = 1 \) for all \( \ell \geq 1 \). We can conclude that, for monomials over \( \ell \) Boolean variables, the minimum margin is at least \( 1/(4\ell) \), resp. at least \( 1/(2\ell) \) as it had been observed by Forster et al. [37].

**Discrete Axis-parallel Hyper-rectangles.** Let \( J(\ell) \in \mathbb{N} \) be polynomially bounded in \( \ell \), and let \( \mathcal{X}_\ell = \{1, \ldots, J(\ell)\}^\ell \). An axis-parallel hyper-rectangle over \( \mathcal{X}_\ell \), also called “\( \ell \)-dimensional box”, is a subset of the form

\[ \{a_1, \ldots, b_1\} \times \cdots \times \{a_\ell, \ldots, b_\ell\} . \]
where \( a_i, b_i \in \{1, \ldots, J(\ell)\} \). Using the predicates \( x_i \geq a \) and \( x_i \leq b \) for \( a, b \in \{1, \ldots, J(\ell)\} \), it is easy to cast an \( \ell \)-dimensional box as a monomial over \( 2\ell J(\ell) \) Boolean variables. It follows that, for boxes of this form, the minimum margin is at least \( 1/(4\ell^2 J(\ell)) \).

**Boolean Decision Lists.** We consider decision lists\(^{14}\) over the Boolean variables \( z_1, \ldots, z_\ell \) of the form

\[
\left[ (z_1^{\sigma_1}, b_1), \ldots, (z_s^{\sigma_s}, b_s), b_{s+1} \right],
\]

where \( 1 \leq s \leq \ell, i(1), \ldots, i(s) \in \{1, \ldots, \ell\}, \) and \( \sigma_1, \ldots, \sigma_s, b_1, \ldots, b_s \in \{0, 1\} \).

A list of this form represents a Boolean function that assigns a bit to each vector \( x \in X_\ell = \{0, 1\}^\ell \) as follows:

- If \( x \) does not satisfy any literal \( z_{i(j)}^{\sigma_j} \) in the list, then return the bit \( b_{s+1} \).
- Otherwise, let \( j \) be smallest index such that \( x \) satisfies the literal \( z_{i(j)}^{\sigma_j} \). Return \( b_j \).

It is well known that Boolean decision lists with a bounded number of label changes (from 0 to 1 or vice versa) within the sequence \( b_1, \ldots, b_s, b_{s+1} \) are linearly separable with polynomially bounded weights. Here, we determine a relatively precise polynomial bound \( P_c(\ell) \) where \( c \) denotes an upper bound on the number of label changes. We view \( c \) as a constant and \( \ell \) as a variable that may grow to infinity. Consider a decision list over the domain \( X_\ell = \{0, 1\}^\ell \) as given in (7.27). By adding redundant items if necessary, we may assume that the list contains every Boolean variable exactly once. For sake of simplicity, we apply a renumbering of indices so that the decision list looks like

\[
\left[ (z_\ell^{\sigma_\ell}, b_\ell), \ldots, (z_1^{\sigma_1}, b_1), b_0 \right].
\]

Let us first assume that \( \sigma_1 = \ldots = \sigma_\ell = 1 \), i.e., all literals in the list are unnegated. We may clearly assume that \( b_1 \neq b_0 \) so that the first label change (from right to left) occurs between \( b_0 \) and \( b_1 \). Let \( c' \leq c \) denote the total number of label changes. Let \( f \) be the Boolean function represented by (7.28). We will design a weight vector \( \mathbf{w}^{[f]} = (w_1^{[f]}, \ldots, w_\ell^{[f]}) \) and a threshold \( t \) such that (7.25) holds. To this end, we decompose \( b_1, \ldots, b_\ell \) (in this order) into \( c' \leq c \) maximal segments without label change. Let \( I_1, \ldots, I_{c'} \) (in this order) denote the corresponding partition of \( \{1, \ldots, \ell\} \) (so that, for instance, \( 1 \in I_1 \) and \( \ell \in I_{c'} \)). Now we choose the weights \( w_1^{[f]}, \ldots, w_\ell^{[f]} \) and the threshold \( t \) for the given decision list according to the following policy:

\(^{14}\)Introduced by Rivest [73].
7.7. Arrangements with a Non-Negligible Minimum Margin

- For all \( j = 1, \ldots, c' \), all weights in \( \{ w_i[f] : i \in I_j \} \) get the same value, say \( \pm W_j \) where \( W_j > 0 \). More precisely, the value is \( W_j \) (resp. \( -W_j \)) if the label associated with the segment \( I_j \) is 1 (resp. 0).

- Let \( W_1 = 1 \) and, for \( j = 1, \ldots, c' \), let \( \ell_j = |I_j| \). The values \( \{ W_j \}_{j=2}^{c'} \) are chosen so that the following holds:

\[
W_j > \ell_{j-1} W_{j-1} + \ell_{j-3} W_{j-3} + \cdots + \begin{cases} 
\ell_1 W_1 & \text{if } j \text{ is even} \\
\ell_2 W_2 & \text{if } j \text{ is odd}
\end{cases}.
\]

- The threshold \( t \) is set to \( 1/2 \) (resp. \( -1/2 \)) if \( b_0 = 0 \) (resp. \( b_0 = 1 \)).

It is easy to verify that any weight vector and threshold chosen in accordance with this policy leads to a representation \( w[f] \) of the function \( f \) associated with (7.28) such that (7.25) holds (provided that \( \sigma_1 = \ldots = \sigma_{\ell} = 1 \)). The following is an easy-to-solve recursion for \( W_j \) leading to weights \( w_1[f], \ldots, w_{\ell}[f] \) that actually do respect the policy:

1. \( \ell_0 = W_0 = 1 \). For \( j = 1, \ldots, c' : \ell_j = |I_j| \).

2. For \( j = 0, \ldots, c' - 1 \): \( W_{j+1} = \sum_{i=0}^{j} \ell_i W_i \).

Suppose that the weights \( W_j \) are chosen according to this recursion. We claim that, for all \( j = 0, 1, \ldots, c' \),

\[
W_j = \sum_{I \subseteq \{1, \ldots, j-1\}} \prod_{i \in I} \ell_i \quad \text{(7.29)}
\]

For \( j = 0 \), this is correct because the right hand-side of (7.29) collapses to \( \prod_{i \in \emptyset} \ell_i = 1 = W_0 \). Assume inductively that the claim is correct for \( W_j \). Then,

\[
W_{j+1} = \ell_j W_j + \sum_{i=0}^{j-1} \ell_i W_i = (\ell_j + 1)W_j
\]

\[
= (\ell_j + 1) \sum_{I \subseteq \{1, \ldots, j-1\}} \prod_{i \in I} \ell_i = \sum_{I \subseteq \{1, \ldots, j\}} \prod_{i \in I} \ell_i,
\]

which proves the claim. The method of Lagrangian multipliers yields that the largest weight \( W_{c'} \), viewed as a function in \( \ell_1, \ldots, \ell_{c'} \) subject to \( c' \leq c \) and \( \sum_{j=1}^{c'} \ell_j = \ell \), is maximized for \( \ell_{c'} = 1 \) and \( \ell_1 = \ldots = \ell_{c'-1} = (\ell - 1)/(c' - 1) \). Thus, we obtain the following upper bound on the weights used for the representation of the decision list:

\[
\sum_{I \subseteq \{1, \ldots, c'-1\}} \prod_{i \in I} \ell_i = \sum_{j=0}^{c'-1} \binom{c'-1}{j} \left( \frac{\ell - 1}{c' - 1} \right)^j
\]

91
Calculus yields that the worst-case is obtained for $c' = c$. We finally remove the artificial assumption $\sigma_1 = \ldots = \sigma_\ell = 1$. Suppose that the parameters $\sigma_i \in \{0, 1\}$ are not necessarily set to $1$. If $\sigma_i = 0$, we should think of $x_i$ in (7.25) as being replaced by $1 - x_i$. Multiplying out, we see that this transformation has an effect on the threshold $t$ only, but not on the weight vector $w^{[f]}$. Thus, the upper bounds obtained for the absolute values of the weights remain valid. The whole discussion can be summarized as follows.

**Theorem 7.18.** Let $c \geq 1$ be a constant. Then, for all $\ell \geq 1$, the following holds. A decision list with at most $\ell$ Boolean variables and with at most $c$ label changes is linearly separable with weights whose absolute values are bounded by

$$P_c(\ell) = \left(1 + \frac{\ell - 1}{c - 1}\right)^{c-1}. $$

### 7.8 Efficiency Issues

Theorem 7.2 ignores efficiency issues, i.e., whether or not the proposed randomized response scheme can be implemented in polynomial time in the relevant parameters of the problem. However, an inspection of its proof in Section 7.6 shows that efficiency is actually obtained under the following conditions:

1. Given $x \in X_\ell$ (resp. $f \in F_\ell$), the vector $u^{[x]}$ (resp. $v^{[f]}$) of the corresponding (low-dimensional) linear arrangement $A_\ell$ can be computed efficiently.

2. Given $f \in F_\ell$, a random instance in $X_{f,0}$ (resp. $X_{f,1}$) can be efficiently generated.

The only little change in the computation of $\tilde{\omega}$ (in comparison to the computation described in Section 7.6) is as follows. We replace the right hand-side of (7.16) by the corresponding empirical average taken over a sample drawn uniformly at random from $X_{f,b}$. An easy application of Hoeffding’s bound shows that (under the assumptions made in Theorem 7.2) the size of this sample can be polynomially bounded in terms of the relevant parameters.

It is straightforward to see that the first condition listed above is satisfied by all classes discussed in Section 7.7, as computing $u^{[x]}$ and $v^{[f]}$ according to (7.26) can be done efficiently. It can also be proved that the same holds for the second condition, as we show next.
Efficient Sampling for Boolean Monomials. Let $\mathcal{F}_\ell$ be the class of Boolean monomials over $\mathcal{X}_\ell = \{0, 1\}^\ell$. For $I \subseteq \{1, 2, \ldots, \ell\}$ and $\sigma \in \{0, 1\}^{|I|}$, let $f(x) = f_I, \sigma(x) = \bigwedge_{i \in I} x_i^{\sigma_i}$ be the corresponding Boolean monomial. A random instance in $\mathcal{X}_{f,1}$ can be generated as follows. For every $i \in I$, set $x_i := \sigma_i$. Then, for every $j \in \{1, 2, \ldots, \ell\} \setminus I$, let $x_j \sim \mathcal{B}$, where $\mathcal{B}$ denote the Bernoulli distribution with parameter $1/2$. On the other hand, a random instance in $\mathcal{X}_{f,0}$ can be generated as follows. Assume $I = \{i_1, \ldots, i_m\}$. Generate a random integer $k \in \{1, 2, \ldots, 2^m - 1\}$ (which can be done efficiently in $m$) and consider its binary representation $\text{bin}(k) = (b_1, \ldots, b_m) \in \{0, 1\}^m$. Note that there exists an index $j \in \{1, 2, \ldots, m\}$ such that $b_j \neq 0$. For every $j \in \{1, 2, \ldots, m\}$, set

$$x_{ij} := \begin{cases} \sigma_{ij} & \text{if } b_j = 0 \\ \neg \sigma_{ij} & \text{otherwise} \end{cases},$$

and draw every other component uniformly at random from $\mathcal{B}$.

Efficient Sampling for Discrete Axis-parallel Hyper-rectangles. Since discrete axis-parallel hyper-rectangles can be cast as monomials over $2\ell J(\ell)$ Boolean variables (as observed in Section 7.7), the result of the previous paragraph applies.

Efficient Sampling for Boolean Decision Lists. As in Section 7.7, we consider Boolean decision lists with a bounded number of label changes. Let $c$ be a constant that bounds from above the number of label changes in the sequence $b_1, \ldots, b_{s+1}$ in (7.27). Using a similar notation as in Section 7.7, we decompose $b_1, \ldots, b_{s+1}$ into $c' \leq c$ maximal segments without label change. Let $I_1, \ldots, I_{c'}$ denote the corresponding partition of $\{1, \ldots, s+1\}$. To each such block we associate a bit, so that $\kappa_i = 1$ denotes that at least one of the literals in the block $I_i$ is satisfied, and $\kappa_i = 0$ denotes that no literal in $I_i$ is satisfied. Moreover, the length of the block $I_i$ is denoted by $\text{len}(I_i)$.

Assume we wish to sample uniformly at random from $\mathcal{X}_{f,1}$ (the case for $\mathcal{X}_{f,0}$ is solved similarly). First, we generate all configurations $\kappa \in \{0, 1\}^{c'}$ for which the bit 1 is returned as output, i.e., all possible truth values for the blocks $I_1, \ldots, I_{c'}$ that lead to the output 1 of the corresponding decision list $f$. Let $T_{f,1}$ be the set containing such configurations. Note that there are at most $2^{c'} \leq 2^c$ configurations in $T_{f,1}$. Therefore, the generation and the corresponding check can be run efficiently. To each configuration $\kappa$ in $T_{f,1}$ we associate the probability

$$\Pr(\kappa) \propto \prod_{i=1}^{c'} p(\kappa_i),$$

(7.30)
where \( p(\kappa_i) \) is defined as follows:

\[
p(\kappa_i) = \begin{cases} 
1 - 2^{-\text{len}(I_i)} & \text{if } \kappa_i = 1 \\
2^{-\text{len}(I_i)} & \text{otherwise}
\end{cases}
\]

Note that (7.30) corresponds to the conditional probability of a configuration \( \kappa \) given an instance \( x \) drawn uniformly at random from \( X_{f,1} \). This probability distribution on \( T_{f,1} \) allows us to efficiently draw (using, for instance, the corresponding cumulative function) a configuration \( \kappa \) for which the output 1 is returned. Once the configuration is fixed, for every \( j = 1, \ldots, c' \), the variables whose indices belong to the block \( I_j \) are generated according to the value of \( \kappa_j \). More specifically, if \( \kappa_j = 0 \) the variables with indices in \( I_j \) are set in a way such that all corresponding literals are not satisfied. If \( \kappa_j = 1 \), we may adopt the technique used in the previous paragraph for Boolean monomials, i.e., we choose a random integer in \( \{1, \ldots, 2^{\text{len}(I_j)} - 1\} \) and exploit its binary representation for setting the variables accordingly (so that at least one of the corresponding literal is satisfied). All the variables which do not appear in the decision list \( f \) are drawn according to \( B \).

### 7.9 Related Work

A first characterization of non-interactive private data analysis in the “centralized” model, where the sensitive database is managed by a trusted curator, is provided by Blum et al. [14]. Specifically, the authors show that, ignoring computational constraints, it is possible to privately release sanitized databases so as to provide \((\alpha, \beta)\)-usefulness, i.e., for all possible choices of \( d \), the answers provided by the mechanism are close (in probability) to the correct ones, for any concept class with polynomial VC-dimension.\(^{15}\)

Private learning in the absence of a trusted curator is addressed by a series of works (see for example [54, 84]). In particular, Kasiviswanathan et al. [54] prove that a concept class is learnable in the statistical query model if and only if it is learnable by what they call a local algorithm. More specifically, they show that there is an efficient mutual simulation between the respective oracles for these models: the SQ-oracle and the so-called LR-oracle. A non-adaptive local algorithm (in the sense of [54]) corresponds to an algorithm which has access to the data via a randomized response scheme in the sense of Definition 7.1. In what follows, we point out the main differences between the achievements of Kasiviswanathan et al. [54] and the main results in this chapter.

\(^{15}\)The VC-dimension [83] is a combinatorial parameter which characterizes the sample complexity for PAC-learning.
• The SQ-learner (resp. the equivalent local algorithm) gets statistical information about a database consisting of data which are labeled according to an unknown target function $f^*$. In combination with the result presented in Theorem 2.6, this offers the possibility to design a non-adaptive SQ-learner (resp. a non-adaptive local algorithm) which weakly learns $(\mathcal{F}_\ell)_{\ell \geq 1}$ under the uniform distribution provided that the SQ-dimension of $\mathcal{F}_\ell$ grows polynomially with $\ell$ only.

• In our setting, we have a known query function $f$ in the role of the unknown target function, but the randomized response scheme is applied to unlabeled data. It should be noted that, even if the SQ-dimension of $\mathcal{F}_\ell$ is polynomially bounded, there is no direct way to transform a given non-adaptive SQ-learner into a successful randomized response scheme of the form (7.1) in our setting. The transformation used by Kasiviswanathan et al. [54] cannot be adopted here because it assumes a labeled database. The main problem in our setting (where $d = (d_1, \ldots, d_n)$ consists of unlabeled data) is to find a single random map $R$ such that the tuple $\mathcal{M}(d) = (R(d_1), \ldots, R(d_n))$, despite having low sensitivity, contains enough information about the labels of all query functions $f \in \mathcal{F}_\ell$. The key for solving this problem is to use a linear arrangement for $\mathcal{F}_\ell$, as we show in Section 7.6.

Gupta et al. [40] show that if a database access mechanism can only access the database by means of the SQ-oracle (or, equivalently, by means of the LR-oracle), then the concept class can be released so as to provide $\varepsilon$-differential privacy and $(\alpha, \beta)$-usefulness if and only if it is agnostically SQ-learnable. A class $\mathcal{F}$ is agnostically SQ-learnable with respect to a distribution $D$ on $X \times \{-1, 1\}$ if, for every $\alpha, \beta > 0$, there exists an SQ-learner $L$ that outputs, with probability at least $1 - \beta$, a hypothesis $h$ such that $Pr_{(Z,b) \sim D}[h(Z) \neq b] \leq Pr_{(Z,b) \sim D}[f^*(Z) \neq b] + \alpha$, where $f^* = \arg\min_{f \in \mathcal{F}} Pr_{(Z,b) \sim D}[f(Z) \neq b]$. In other words, this framework models the problem of learning a class $\mathcal{F}$, agnostic to whether or not any function $f \in \mathcal{F}$ can provide a good labeling of the data.

The main difference between the results of Gupta et al. [40] and the contribution presented in this chapter consists in the notion of usefulness that the randomized response scheme must achieve. The work of Gupta et al. [40] requires the private mechanism to output values which are $\alpha$-close to the true answers with probability at least $1 - \beta$ for all possible choices of the input database $d$, while we are satisfied if this property holds for $(f, \omega, n)$-random databases. In other words, we have replaced the worst-case analysis of Gupta et al. [40] by a kind of average-case analysis. While the worst-case analysis establishes the very high barrier of “agnostic SQ-learnability”, our main results show that the barrier in the average case, namely weak SQ-learnability under the uniform distribution, is much lower.
Chapter 7. Randomized Response Schemes and the SQ-model

7.10 Conclusions

We investigated to what extent randomized response schemes can combine $\varepsilon$-differential privacy with the conflicting goal of providing useful answers to counting queries taken from a known class $\mathcal{F}$. We introduced the notions of weak and strong usefulness and proved the following results. First, if $\mathcal{F}$ cannot be weakly SQ-learned under the uniform distribution, then $\varepsilon$-differential privacy rules out weak usefulness. Second, for a broad variety of classes $\mathcal{F}$ that actually can be weakly SQ-learned under the uniform distribution, we designed a randomized response scheme that provides $\varepsilon$-differential privacy and strong usefulness.
Bibliography


List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathds{1}_S(x)$</td>
<td>Characteristic function of a set $S$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Accuracy parameter</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Confidence parameter (accuracy)</td>
</tr>
<tr>
<td>$\bar{\gamma}_{\text{min}}(A)^{-1}$</td>
<td>Margin complexity of a sign matrix $A$</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Approximation parameter (privacy)</td>
</tr>
<tr>
<td>$\Delta(g)$</td>
<td>Sensitivity of a function $g$</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>Privacy parameter</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Partition of an integer $n$</td>
</tr>
<tr>
<td>$A$</td>
<td>Linear arrangement for a sign matrix</td>
</tr>
<tr>
<td>$\mathcal{B}$</td>
<td>Bernoulli distribution with parameter $1/2$</td>
</tr>
<tr>
<td>$b_{\nu,k}^{(H)}$</td>
<td>Bernstein basis polynomials of order $H$ and degree $k$</td>
</tr>
<tr>
<td>$B_k^{(H)}(g; y)$</td>
<td>Iterated Bernstein polynomial of a function $g$ of order $H$ and degree $k$ (evaluated at $y$)</td>
</tr>
<tr>
<td>$\mathcal{D}$</td>
<td>Distribution over $\mathcal{X}$</td>
</tr>
<tr>
<td>$d$</td>
<td>Database over $\mathcal{X}$</td>
</tr>
<tr>
<td>$d \approx d'$</td>
<td>Neighboring databases</td>
</tr>
<tr>
<td>$\text{diag}(v)$</td>
<td>Matrix with the components of $v$ on the main diagonal and zero elsewhere</td>
</tr>
<tr>
<td>$\text{err}_D(h, f^*)$</td>
<td>Error of $h \in \mathcal{H}$ with respect to $f^* \in \mathcal{F}$ and distribution $\mathcal{D}$</td>
</tr>
</tbody>
</table>
**List of Symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX($f^*$, $D$)</td>
<td>Example oracle</td>
</tr>
<tr>
<td>$\mathcal{F}$</td>
<td>Concept class</td>
</tr>
<tr>
<td>$f$</td>
<td>Concept from $\mathcal{F}$</td>
</tr>
<tr>
<td>$f^*$</td>
<td>Target concept from $\mathcal{F}$</td>
</tr>
<tr>
<td>FB($A$)</td>
<td>Forster bound for a sign matrix $A$</td>
</tr>
<tr>
<td>$\mathcal{G} = (\mathcal{K}, \mathcal{E})$</td>
<td>Graph $\mathcal{G}$, where $\mathcal{K}$ is the set of nodes and $\mathcal{E}$ is the set of edges</td>
</tr>
<tr>
<td>$\mathcal{H}$</td>
<td>Hypothesis class</td>
</tr>
<tr>
<td>$h$</td>
<td>Hypothesis from $\mathcal{H}$</td>
</tr>
<tr>
<td>KLD($P, Q$)</td>
<td>Kullback-Leibler divergence from $Q$ to $P$</td>
</tr>
<tr>
<td>KN</td>
<td>Kernel function</td>
</tr>
<tr>
<td>$\mathcal{L}$</td>
<td>Learning algorithm</td>
</tr>
<tr>
<td>$\mathcal{L}$</td>
<td>Lattice cover</td>
</tr>
<tr>
<td>$\ell$</td>
<td>Complexity parameter or representation size of the elements of $\mathcal{X}$ (resp. $\mathcal{Y}$)</td>
</tr>
<tr>
<td>Lap($\lambda$)$^r$</td>
<td>Zero-mean $r$-dimensional Laplace distribution with parameter $\lambda$</td>
</tr>
<tr>
<td>LD</td>
<td>Laplace probability density function</td>
</tr>
<tr>
<td>LP</td>
<td>Linear program</td>
</tr>
<tr>
<td>LS</td>
<td>Loss function</td>
</tr>
<tr>
<td>$\mathcal{M}$</td>
<td>(Privacy) mechanism</td>
</tr>
<tr>
<td>$\mathcal{N}(\mu, \Sigma)$</td>
<td>(Multivariate) Gaussian distribution of mean $\mu$ and covariance matrix $\Sigma$</td>
</tr>
<tr>
<td>$n$</td>
<td>Size of a database $d$</td>
</tr>
<tr>
<td>obj</td>
<td>Objective function of a linear program</td>
</tr>
<tr>
<td>$\mathcal{P}$</td>
<td>Set of integer partitions</td>
</tr>
<tr>
<td>$\mathcal{R}$</td>
<td>Response universe of a mechanism $\mathcal{M}$</td>
</tr>
<tr>
<td>$\mathcal{R}^*$</td>
<td>Minimax risk</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$R$</td>
<td>Random map employed in a randomized response scheme</td>
</tr>
<tr>
<td>$\text{SD}(P, Q)$</td>
<td>Statistical difference between two probability distributions $P, Q$</td>
</tr>
<tr>
<td>$\text{SQdim}_D(F)$</td>
<td>Statistical query dimension of $F$ with respect to $D$</td>
</tr>
<tr>
<td>$\text{STAT}(f^*, D)$</td>
<td>SQ-oracle</td>
</tr>
<tr>
<td>$\mathcal{U}$</td>
<td>Uniform distribution over $\mathcal{X}$</td>
</tr>
<tr>
<td>$U$</td>
<td>Utility (or score) function</td>
</tr>
<tr>
<td>$\mathcal{X}$</td>
<td>Sensitive data domain</td>
</tr>
<tr>
<td>$\mathcal{Y}$</td>
<td>Set of query points</td>
</tr>
<tr>
<td>$|A|_2$</td>
<td>Frobenious norm of a matrix $A$</td>
</tr>
<tr>
<td>$|A|_2$</td>
<td>Spectral norm of a matrix $A$</td>
</tr>
<tr>
<td>$|v|_1$</td>
<td>Manhattan norm of a vector $v$</td>
</tr>
<tr>
<td>$|v|_2$</td>
<td>Euclidean norm of a vector $v$</td>
</tr>
<tr>
<td>$|v|_{\infty}$</td>
<td>Uniform norm of a vector $v$</td>
</tr>
</tbody>
</table>
List of Figures

6.1 Private KDE with Gaussian kernel . . . . . . . . . . . . . . . . . 61
6.2 Private SVM with Gaussian kernel . . . . . . . . . . . . . . . . . 64
7.1 Randomized response scheme . . . . . . . . . . . . . . . . . . . . . 72