A Study of Privacy and Fairness in Sensitive Data Analysis

Moritz A.W. Hardt

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
By the Department of
Computer Science

Advisor: Boaz Barak

November 2011

Abstract

In this thesis we consider the challenges arising in the design of algorithms that interact with sensitive personal data—such as medical records, online tracking data, or financial records.

One important goal is to protect the privacy of those individuals whose personal information contributed to the data set. We consider algorithms that satisfy the strong privacy guarantee known as *differential privacy*. A wide range of computational tasks reduces to the setting in which a trusted database curator responds to a number of statistical queries posed by an untrusted data analyst. The basic question is how *accurately* and *efficiently* the curator can release approximate answers to the given queries while satisfying differential privacy. We make the following main contributions to differentially private data analysis:

— We expose a connection between differential privacy and certain problems in convex geometry revolving around a deep conjecture known as the Hyperplane conjecture. Assuming the truth of this conjecture we give differentially private mechanisms with nearly optimal accuracy in the case where the queries are given all at once (non-interactively) and the number of queries does not exceed the database size.

— Multiplicative weights mechanisms are a powerful tool in algorithms, machine learning and optimization. We introduce a privacy-preserving multiplicative weights framework suitable for answering a huge number of queries even in the interactive setting. The accuracy of our algorithm in terms of database size and number of queries matches the statistical sampling error that already arises in the absence of any privacy concerns. Our algorithm can also be used to produce a differentially private synthetic data set encoding the curator's answers. For this task the runtime of our algorithm—which is linear in the universe size—is essentially optimal due to a prior cryptographic hardness result.

— We then consider avenues for obtaining differentially private algorithms with a runtime polynomial in the size of the data set or at least subexponential in the universe size. Based on a new learning algorithm for submodular functions, we present the first polynomial-time algorithm for answering a large number of Boolean conjunction queries (or contingency tables) with non-trivial accuracy guarantees. Conjunction queries are a widely used and important class of statistical queries.

— Furthermore, we exhibit an explicit and efficient reduction from the problem of privately releasing a class of queries to the problem of non-

privately learning a related class of concepts. Instantiating this general reduction with new and existing learning algorithms yields several new results for privately releasing conjunctions and other queries.

Not all problems arising in the presence of sensitive data are a matter of privacy. In the final part of this thesis, we isolate *fairness in classification* as a formidable concern and thus initiate its formal study. The goal of fairness is to prevent discrimination against protected subgroups of the population in a classification system. We argue that fairness cannot be achieved by blindness to the attribute we would like to protect. Our main conceptual contribution is in asserting that fairness is achieved when *similar individuals are treated similarly*. Based on the goal of treating similar individuals similarly, we formalize and show how to achieve fairness in classification, given a similarity metric. We also observe that our notion of fairness can be seen as a generalization of differential privacy.

Acknowledgments

I am most grateful to my advisor Boaz Barak. He has been a generous and inspiring advisor throughout. Boaz taught me how important it is to keep an open mind about all research directions rather than drawing artificial borders around research areas. If I have grown as an independent researcher, it was primarily by learning from Boaz. It was not always easy. One of the early lessons I struggled with was how to write the letters *'n'* and *'m'* legibly. I thank Boaz for his great patience with me.

My academic life has been shaped tremendously by three wonderful mentors. I owe a huge debt of gratitude to Cynthia Dwork, Steven Rudich and Kunal Talwar. Steven put a lot of faith in me even before I started graduate school. Our long conversations on complexity theory shaped my view of the field. I aspire to his standards of teaching and writing. Kunal introduced me to differential privacy during an incredibly exciting internship at Microsoft Research. In hindsight, our collaboration was the starting point of this thesis. Cynthia gave me the exciting opportunity to participate in developing a new notion. This has been a fascinating experience. I am also grateful to her for sharing her outstanding advice with me on so many occasions.

I had the pleasure of collaborating with many brilliant scholars over the years. I would like to thank all my co-authors for their support and everything they have taught me. Special thanks to Frank McSherry, Aaron Roth and Guy Rothblum. Thanks also to my entire thesis committee: Sanjeev Arora, Boaz Barak, Moses Charikar, Rob Schapire and Rocco Servedio.

Many thanks to all my friends for making my time in graduate school so much more meaningful and enjoyable. I would like to remember Björn Walter—my first intellectual companion. I dearly miss him.

I thank my fiancée Michaela Götz for her love and her faith in our relationship. I am proud of us for defeating the long distance between us during grad school. In her role as a researcher, Michaela is also a candid critic of my work and has helped to improve it.

Thanks to my mother Petra Hardt, my sister Julia Hardt, and my brother Claudius Hardt, for their love and support over the years. This thesis is written in loving memory of my father, Manfred Hardt.

I acknowledge support through NSF grants CCF-0426582 and CCF-0832797, and, a grant from the Packard foundation.

Midway upon the journey of our life
I found myself within a forest dark,
For the straightforward pathway had been lost.
— Dante Allighieri

Dedicated to the memory of my father

Contents

List of Figures

Chapter 1

Introduction

Technological advances do not always work in the favor of the individual. In the context of *privacy*, this insight was first articulated in the seminal article "The Right to Privacy" from 1890 by Warren and Brandeis [WB]. At the time, an increasing number of personal photographs of public figures appeared in newspapers due to cheaper photography and printing devices. Alarmed by this trend, Warren and Brandeis argued that every individual shall have the right to privacy which they loosely defined as the "right to be let alone."

More than one-hundred twenty years later our understanding of privacy and our methods of ensuring privacy are still quite limited. The ubiquity of computers, personal electronics, and the internet have given rise to an abundance of ways to collect and publish personal data. A rapidly increasing stream of commercial and scientific applications incentivizes the analysis of sensitive data sets. Failure to protect privacy may result in significant harm to the individual and to society.

The field of *privacy-preserving data analysis* today includes many scientific disciplines such as statistics and machine learning, theoretical computer science, cryptography, security and databases. A variety of computational tasks that a data analyst might wish to carry out reduce to the problem of obtaining accurate statistics about the data set. The central question in privacy-preserving data analysis can be summarized as:

> **Question 1:** How can we release *accurate* statistics about a data set while protecting the *privacy* of those individuals who contributed their data?

Motivating examples are medical studies, such as the Genome Wide Association Studies (GWAS) conducted by the National Institute of Health (NIH), an agency of the U.S. Department of Health & Human Services. Genome Wide Association Studies aim at discovering the association between human genes and common diseases such as cancer in order to aid the development of better treatments. In a typical study the NIH produces aggregate allele counts of a case group (patients with the disease) and a control group (individuals without the disease). Unfortunately, the NIH had to shut down public access to its data sets after research pointed at a significant privacy risk for the participants

of GWAS. Specifically, Homer et al. [HSR+, GCN+] showed that participation of a specific individual in the case group of a study can be determined from the individual's DNA and the published allele counts—thus revealing the fact that the individual has the disease common to all participants in the case group.

The case of GWAS illustrates several important points that are recurrent throughout a long history of privacy pitfalls. In particular, a privacy breach occurred even though:

— No computer system was compromised or malfunctioned.

— The motives of the data curator (in this case the National Institute of Health) were pure and the data curator was trustworthy.

— The data released, here, a collection of aggregated allele counts, appeared innocuous and harmless. In particular, the data did not contain any information obviously identifying an individual, such as names.

GWAS is only one example in a growing line of privacy failures with similar characteristics, such as the re-identification of users from "anonymized" Massachusetts medical records [Swe], AOL search logs [BZ], or, the Netflix prize data (see [NS]).

Another important common characteristic of many privacy breaches is that the privacy measures taken in each case fail to account for the existence of *auxiliary information* available to the attacker. Indeed, the attack described in [HSR+] on the GWAS data uses rich background information about the human genome available in the public domain. Auxiliary information also played a key role when Netflix, an online movie rental service, released the anonymous movie ratings of 500,000 subscribers of Netflix. Netflix announced a $1,000,000 prize to whomever could improve their recommendation system by 10% based on the released data set. Narayanan and Shmatikov showed how to re-identify anonymous users in the data set by matching their movie ratings with reviews given by users on IMDB, a large online movie data base [NS]. A second Netflix challenge was cancelled after the Federal Trade Commission raised privacy concerns [Hun2].

These examples demonstrate the difficulty of balancing the privacy needs of individual participants in a data set and the utility of publishing statistics about the data set. What makes Question 1 so challenging to begin with is the problem of *how* one should define the notion of *privacy*. What is meant by *utility* (or *usefulness*) is often less subtle and arises naturally in application settings.

1.1 Differential Privacy

In this thesis we consider the privacy guarantee known as *differential privacy*. Differential privacy is a formal notion due to Dwork et al. [Dwo1, DMNS] that attaches a rigorous meaning to the word *privacy*. Intuitively speaking, differential privacy gives the strong guarantee that:

> The presence or absence of any single individual in a data set will only insignificantly affect the outcome of an analysis.

More precisely, differentially private algorithms are randomized algorithms whose output distribution remains nearly unchanged even if we perturb the information of a single participant arbitrarily. It is easy to see that randomization is necessary. Indeed, the algorithm must be able to produce the same output on different data sets. Unless the algorithm gives always the same output (thereby defeating any notion of utility), randomization becomes essential.

Differential privacy helps to incentivize the participation of an individual in a data set. That is because the risk resulting from participation of an individual in a database is not significantly greater than that of opting out. This type of guarantee is not to be confused with the stronger statement that "nothing about an individual should be learnable from the database that cannot be learned without access to the database". Such a desideratum was articulated by Dalenius [Dal], but Dwork [Dwo1] showed a general impossibility result proving that this goal cannot be achieved.

In light of our discussion above, differential privacy has the appealing feature that it provides a privacy guarantee while making very mild assumptions on the background knowledge of the adversary. Differential privacy is thus highly resilient to attacks utilizing unanticipated auxiliary information. But see [KM2] for settings in which care must be taken when applying differential privacy.

There are several other useful properties that make differential privacy a robust definition. For instance, differential privacy *composes* gracefully in the sense that the interaction of two differentially private mechanisms remains differentially private up to a small quantitative loss in the privacy guarantee (not expressed in the informal description above). We refer the reader to surveys [Dwo2, Dwo3] for additional motivation of the definition.

As differential privacy poses a strong requirement, it becomes a challenging task to design useful algorithms that satisfy differential privacy. The most basic and well-studied setting of differential privacy is the case where a trusted database curator responds to a number of queries given by an (untrusted) data analyst. The queries that the analyst may ask are so-called *statistical queries*.

This includes several important classes of queries such as *counting queries* and *contingency tables*; many learning algorithms can be implemented using statistical queries [BDMN].

The requirement is that the answers of the database curator satisfy differential privacy. It is easy to satisfy this requirement by answering each query with random noise independent of the data set. The fundamental trade-off we study is therefore between *accuracy* and differential privacy:

> **Question I':** How accurately can we answer a set of statistical queries while maintaining differential privacy?

Accuracy is not the only important concern in the design of differentially private algorithms. Often the "price" of ensuring differential privacy is paid in terms of *computational inefficiency*. A case in point is the fundamental work of Blum, Ligett and Roth [BLR] showing that in absence of computational constraints one can obtain a differentially private data structure encoding answers to a huge number of statistical queries with non-trivial accuracy. However, the running time of the algorithm was super-polynomial in the universe from which the data is drawn. The size of the universe could itself be exponential in the database size (the natural input size). Without a privacy constraint, the same task requires no effort as outputting data set itself is sufficient. This motivates the following important question:

> **Question II:** Can we reduce the computational gap between differentially private and non-private data analysis?

In this thesis we provide several new positive answers—and some negative answers—to both questions. Our contributions are outlined next.

1.2 Contributions to Differential Privacy

1.2.1 The geometry of Differential Privacy

In Chapter 3 of this thesis we will translate Question I' into a purely geometric question. In particular, we appeal to a deep conjecture from convex geometry known as the *Hyperplane conjecture* [Bou, MP, Kla] as well as recent results on the volume and isotropic constant of high-dimensional convex bodies [LPRTJ, KK].

Assuming the truth of the Hyperplane conjecture, we provide an *optimal* answer to the question in the case where the queries are given all at once (non-interactively) and the number of queries does not exceed the database size. The previously best known mechanism in this setting was the Laplace

mechanism [DMNS]—the most basic and widely used differentially private mechanism.

To give an optimal trade-off, we first show a new unconditional lower bound on the error of any differentially private mechanism. Our lower bound is the first result that separates differential privacy from a well-studied weakening of differential privacy. We then give a new algorithm matching the lower bound. Roughly speaking our algorithm achieves error $O(\sqrt{k})$ per query when given k queries, while the Laplace mechanism achieves $O(k)$. Perhaps even more importantly, our work exhibits the *geometric* nature of the problem leading to a range of new techniques.

This result is joint work with Kunal Talwar and first appeared in the Symposium on the Theory of Computing (STOC 2010) [HT].

1.2.2 Privacy-preserving multiplicative weights framework

In several applications, the number of queries may grow significantly beyond the size of the data set. In this case the methods described above no longer give meaningful accuracy. The beautiful work of Blum, Ligett and Roth [BLR] showed that even when the number of queries is huge compared to the size of the data set non-trivial accuracy is possible. Two shortcomings of this work were that the runtime of the algorithm had a super-polynomial dependence on the size of the universe (from which the data is drawn) and the queries had to be asked non-interactively.

In Chapter 4 we address both of these shortcomings. In particular, we develop a differentially private multiplicative weights mechanism that can handle a huge number of statistical queries interactively with a runtime that grows only *linearly* with the universe size. The accuracy of our mechanism is $O(\sqrt{n \log k})$ on a data set of size n and k queries. This bound matches the so-called statistical sampling error that arises already in the absence of any privacy constraints. In terms of both accuracy and runtime we improve upon a long line of work [BLR, DNR+, DRV, RR].

Our algorithm can also be used to produce differentially private *synthetic data* and for this task our runtime is essentially best possible under cryptographic assumptions [DNR+, UV].

On an intuitive level our algorithm uses the multiplicative weights framework to *learn* a differentially private approximation to the true data set. Multiplicative weights mechanisms are powerful tools in algorithms design, machine learning, and even complexity theory (see, e.g., [AHK]). Our work enables the use of some of this technology in the context of differential privacy.

In Chapter 5, we demonstrate that the multiplicative weights framework

leads to a simple and practical release mechanism in the non-interactive setting. We demonstrate the practicality of our approach by evaluating the algorithm on several data sets where differential privacy previously failed (see, e.g., [FRY]).

The results in Chapter 4 are joint work with Guy Rothblum and appeared in the Foundations of Computer Science (FOCS 2010) [HR]. Chapter 5 is joint work with Frank McSherry and Katrina Ligett [HLM].

1.2.3 Connections to learning theory

In an ideal world, an algorithm for ensuring differential privacy should be as efficient as the computations that the analyst intended to perform on the data in the first place. Machine learning captures a broad and useful class of data analysis tools. Unfortunately, there is presently a huge gap in complexity between machine learning and private data analysis. Efficient learning algorithms usually depend only logarithmically on the size of the universe from which the data is drawn. In contrast, the state of the art algorithms in differential privacy described earlier have a *linear* dependency on the size of the universe. This leads to the following refinement of Question II:

> **Question II':** Can we reduce the *complexity gap* between machine learning and differentially private data analysis?

To approach this question, we consider the problem of answering Boolean conjunctive queries over a data set $D \subseteq \{0,1\}^d$ (the universe here is $\{0,1\}^d$). Conjunctions (and the closely related notion of *contingency tables* or *data cubes*) are an example of statistical queries that are particularly well-studied and relevant in practice. For example, the U.S. Census Bureau uses contingency tables to release statistical information. See, e.g., [BCD+, KRSU] and the references therein for further motivation.

In Chapter 6 we give the first algorithm with runtime polynomial in d that outputs a data structure which encodes differentially private answers to 99% of all width-w Boolean conjunctions for any width $w \in \{1,\ldots,d\}$ up to an error of $|D|/100$. Our result turns out to be more general and applies to any set of queries that can be described by a *submodular* function. This includes many natural classes of queries, such as the cut function of a graph (e.g., a social network). Implicit in our result is a reduction from the problem of privately releasing conjunctions to learning submodular functions. The main technical ingredient is new learning algorithm for submodular functions inspired by the recent work of [BH].

More generally, we ask: *How many statistical queries to a data set are needed to learn approximate answers to all queries from a certain concept class?* Using the

multiplicative weights approach from Chapter 4, we show that the answer is essentially equal to the *agnostic learning* query complexity of the same concept class in Kearns' statistical query model [Kea]. Using existing lower bounds on the agnostic query complexity [BFJ+, Fel], this result gives unconditional lower bounds on the complexity of any differentially private mechanism that can be implemented in the statistical query model. This includes almost all known algorithms in differential privacy.

Chapter 6 is joint work with Anupam Gupta, Aaron Roth and Jonathan Ullman and appeared in the Symposium on the Theory of Computing (STOC 2011) [GHRU].

An explicit reduction from private data analysis to learning theory

An appealing approach toward Question II' is to exhibit *efficient* reductions from private data analysis to problems in learning theory. In Chapter 7 we demonstrate the viability of this approach by giving an explicit and efficient reduction from the problem of privately releasing a query class $\mathcal{Q}$ to the problem of non-privately *learning* sums of thresholds over $\mathcal{Q}$. We instantiate this general reduction with a variety of algorithms for learning thresholds (mainly based on [Jac, JKS, KS]). These instantiations yield several new results for differentially private data release. As an example, taking $\{0,1\}^d$ to be the data domain (of dimension d), we obtain differentially private algorithms for releasing all width-w conjunction queries (or w-way contingency tables). For any given w, the resulting data release algorithm has bounded error as long as the database is of size at least $d^{O\left(\sqrt{w \log(w \log d)}\right)}$ (ignoring the dependence on other parameters). The running time is polynomial in the database size. The best sub-exponential time algorithms known prior to our work required a database of size $\widetilde{O}(d^{w/2})$ due to [DMNS].

Chapter 7 is joint work with Guy Rothblum and Rocco Servedio and is to appear in the Symposium on Discrete Algorithms (SODA 2012).

1.3 Fairness in classification

Not all problems arising in the presence of sensitive data sets are a matter of privacy. In the final chapter of this thesis we turn to the problem of *fairness in classification*. Nearly all classification tasks face the challenge of achieving utility in classification for some purpose, while at the same time preventing discrimination against protected population subgroups.

Consider the example of an *advertising network*, such as Google's Adsense network. An advertising network collects and maintains information about a set of individual users and serves certain advertisements to individuals

based on characteristics of the individual. The promise of online targeted advertising is to accurately decide which ad to deliver to a user based on tracking information about the user such as their browsing history. The classifier in this example is the algorithm that decides whether a particular advertisement is shown to a given user. An emerging concern in online targeted advertising is the danger that an advertiser could knowingly or unknowingly target individuals based on protected attributes such as race, religion or medical conditions. This could result in steering minorities into less advantageous offers as discussed in a recent article in The Wall Street Journal [SA1].

The underlying problem here is restricted neither to advertising not the online world, but arises in such diverse areas as high school admissions, health care [SM], banking etc. Broadly speaking, the question we ask is:

> **Question III:** How can we prevent discrimination against protected population subgroups in classification systems?

One benefit of preventing such discrimination is that individuals are treated fairly. But there can also be important benefits to the party doing the classification (the advertiser, lender, admissions officer, *etc*.) such as freedom from regulatory concern.

In Chapter 8, we initiate the formal study of fairness in classification. We argue that fairness cannot be obtained by blindness to the attribute we would like to protect, but rather requires some understanding of the correlations implied by this attribute. We further argue that "fairness on average" towards the entire protected group is insufficient and could be seriously abused. The main conceptual contribution of this chapter is in asserting that fairness is achieved when *similar individuals are treated similarly*. We argue that understanding the degree to which individuals from different groups are similar with respect to a certain classification task requires an understanding of the cultures of the two groups. It is our contention that similarity metrics are applied in many contexts, but these are often hidden. Our work explicitly exposes the metric, opening it to public discussion and debate.

Based on the goal of treating similar individuals similarly, we formalize and show how to achieve fairness and utility in classification, given a similarity metric. Specifically, our *local* notion of fairness requires that the classification algorithm is randomized and that the classification of any two similar individuals (with respect to the given metric) results in two statistically close distributions over outcomes. At this high-level there is a close connection between fairness and differential privacy. Recall, differential privacy requires an algorithm to produce statistically close distributions on any two databases that are "similar" in the sense that they differ only in one individual. We exploit this conceptual analogy by transferring some of the techniques from

differential privacy to the fairness setting. This leads to a quantitative trade-off between utility and fairness under a natural assumption on the metric space known as bounded doubling dimension.

We also give conditions on the metric under which our local notion of fairness implies *statistical parity*, which says that the demographics of the accepted group is the same as the demographics of the underlying population. In a complementary setting, we propose tools for what can be viewed as "fair affirmative action." Namely, we give methods for guaranteeing statistical parity for a group while treating similar individuals as similarly as possible.

Chapter 8 is joint work with Cynthia Dwork, Toniann Pitassi, Omer Reingold and Richard Zemel.

Chapter 2

Background

In this chapter we present the background material that we rely on throughout this thesis. We start with the formal definition of differential privacy in the next section. We then continue with a survey of prior work and relevant techniques.

In what follows we let $\log(x)$ denote the natural logarithm and $\exp(x) = e^x$ denotes the exponential function. When $v \in \mathbb{R}^k$ is a real-valued vector we denote by $\|v\|_p$ its ℓ_p-norm.

2.1 Databases and differential privacy

We consider a finite data universe $\mathscr{U}$. An element $u \in \mathscr{U}$ will be sometimes be called a *data item*. A *data set* or *database* is formally a multiset $D\colon \mathscr{U} \to \mathbb{N}$, where $\mathbb{N} = \{0, 1, 2, \dots\}$ is the set of natural numbers including 0. The *size* of the data set is defined as $|D| = \sum_{u \in \mathscr{U}} D(u)$. The set of all finite databases is denoted by $\mathscr{D} \stackrel{\text{def}}{=} \mathscr{U} \to \mathbb{N}$ and we put $\mathscr{D}_n = \{D \in \mathscr{D} : |D| = n\}$.

For example, in the context of a medical study, the data universe $\mathscr{U}$ most naturally corresponds to the set of all possible responses to the study. The database stores for each participant the corresponding response.

Definition 2.1.1. Two data sets $D, D' \in \mathscr{D}$ are called *adjacent* or *neighboring* if

$$\|D - D'\|_1 \stackrel{\text{def}}{=} \sum_{u \in \mathscr{U}} \left|D(u) - D'(u)\right| \leqslant 1\,.$$

That is, D and D' differ in the value of at most 1 individual.

Intuitively, D and D' are neighboring databases if D' was obtained from D through the "opting out" or "misreporting" of a single individual.

We will be interested in mappings from $\mathscr{D}$ into the set of probability measures over some abstract range $\mathscr{R}$. We typically think of $\mathscr{R}$ as the set of possible outcome of some operation on the database. We let $\mu(\mathscr{R})$ denote the set of all probability measures over $\mathscr{R}$.

A *mechanism* is a mapping $M\colon \mathscr{D} \to \mu(\mathscr{R})$. Alternatively, we sometimes think of a mechanism as a collection of probability measures $M = \{\mu_D : D \in$

$\mathscr{D}$}. Intuitively speaking, a mechanism is a randomized algorithm accessing the database and implementing some functionality. We will be interested in mechanisms that satisfy differential privacy.

Definition 2.1.2 (Differential Privacy [DMNS]). A mechanism $M : \mathscr{D} \to \mu(\mathscr{R})$ satisfies (ε, δ)-*differential privacy* if for all $S \subseteq \mathscr{R}$ and every pair of two adjacent databases D, D', we have

$$\mathbb{P}\{M(D) \in S\} \leqslant e^{\varepsilon}\, \mathbb{P}\{M(D') \in S\} + \delta. \tag{2.1}$$

If $\delta = 0$, we say the algorithm satisfies ε-*differential privacy*.

Note that for small ε, we have $e^{\varepsilon} \approx 1 + \varepsilon$. The choice of e^{ε} makes the definition mathematically more well-behaved. Definition 2.1.2 formalizes the intuition explained in the introduction. Indeed, the participation of a single individual in the data set changes the probability distribution of any differentially private mechanism only very slightly in the sense expressed in Equation 2.1. Sometimes it is helpful to work with a condition expressed in the next lemma in place of (ε, δ)-differential privacy.

Lemma 2.1.3. *Suppose a mechanism $M : \mathscr{D} \to \mu(\mathscr{R})$ satisfies for all adjacent D, D' with $P = M(D), Q = M(D')$ that*

$$\mathbb{P}_{v \sim P}\left\{\left|\log\left(\frac{P(v)}{Q(v)}\right)\right| > \varepsilon\right\} \leqslant \delta. \tag{2.2}$$

Then, M satisfies (ε, δ)-differential privacy.

Proof. Indeed, suppose (2.2) is satisfied and consider $B = \{v : |\log(P(v)/Q(v))| > \varepsilon\}$. Let $S \subseteq \mathscr{R}$ and consider $S_1 = S \cap B$ and $S_2 = S \cap B^c$. We then know that

$$P(S) = P(S_1) + P(S_2) \leqslant \delta + e^{\varepsilon} Q(S_2) \leqslant e^{\varepsilon} Q(S) + \delta. \qquad \blacksquare$$

In the lemma above and throughout this thesis we use the notation $v \sim P$ to indicate that v is a random variable drawn according to the distribution P. We will sometimes put random variables in boldface font to avoid confusion.

2.1.1 Histogram representation

Often it will be convenient to view data sets in terms of their *histogram* representation. A histogram is just the natural vector representation of a data set. Formally, a *histogram* over a universe $\mathscr{U}$ is a vector $x \in \mathbb{N}^{\mathscr{U}}$. Typically we will identify $\mathscr{U}$ with the set $[N] \stackrel{\text{def}}{=} \{1, \ldots, N\}$ for some natural number N. In this case, $x \in \mathbb{N}^N$. A histogram x represents a data set D if for all data items

$u \in \mathcal{U}$ we have $x_u = D(u)$. Note that two data sets D, D' are adjacent if and only if their corresponding histograms x, x' satisfy $\|x - x'\|_1 \leqslant 1$.

We can extend the notion of a histogram to that of a *fractional histogram* which is simply a vector $x \in \mathbb{R}_+^N$ where $\mathbb{R}_+$ is the set of non-negative real numbers. The definition of (ε, δ)-differential privacy extends naturally to mechanisms $M \colon \mathbb{R}^N \to \mu(\mathcal{R})$ by requiring that Condition 2.1 be satisfied for all $x, x' \in \mathbb{R}^N$ such that $\|x - x'\|_1 \leqslant 1$.

Normalized histograms. Sometimes we will consider *normalized* histograms. In this case we assume that $x \in \mathbb{R}_+^N$ satisfies $\sum_{u \in \mathcal{U}} x_u = 1$. This is often convenient as we may then think of x as specifying a distribution over the universe. In this case, differential privacy must be satisfied with respect to all $x, x \in \mathbb{R}_+^N$ that satisfy $\|x - x'\|_1 \leqslant 1/n$.

2.2 Queries and sensitivity

A *query* is a mapping $q \colon \mathcal{D} \to \mathcal{R}$. In a typical setting $\mathcal{R} = \mathbb{R}^k$ for some $k \geqslant 0$. In this case we think of q as making k numerical queries about the data set. A *class of queries* is a set $\mathcal{Q} \subseteq \mathcal{D} \to \mathcal{R}$.

An important parameter of a query is its *sensitivity* defined next.

Definition 2.2.1 (Sensitivity). The ℓ_1*-sensitivity* of a query $q \colon \mathcal{D} \to \mathbb{R}^k$ is defined as

$$\Delta(q) \stackrel{\text{def}}{=} \max_{D,D'} \|q(D) - q(D')\|_1 ,$$

where the supremum is taken over all neighboring data sets D, D'.

Later we will be concerned with differentially private mechanisms answering queries. The sensitivity of the query will have an important effect on how much perturbation is required in answering the query. Throughout this work we will mainly consider statistical queries, counting queries and linear queries as defined next. These three classes are all very closely related.

2.2.1 Counting queries, statistical queries, linear queries

A basic class of queries are counting queries. Counting queries allow the data analyst to count how many individuals in the database satisfy a specific predicate (e.g., "how many individuals smoke and have lung cancer?"). Formally, a *counting query* $q \colon \mathcal{D} \to \mathbb{R}$ is specified by a predicate $P \colon \mathcal{U} \to \{0, 1\}$ on the data universe. Its value on a database D is defined as $q(D) = \sum_{u \in \mathcal{U}} P(u) D(u)$. We have $q(D) \in \{0, 1, \ldots, n\}$. With some abuse of notation we will sometimes identify q with the predicate that defines it. For every counting query $q \colon \mathcal{D} \to \mathbb{R}$

we have $\Delta(q) = 1$. We can represent multiple counting queries $q_1, \ldots, q_k$ by a single query $F\colon \mathscr{D} \to \mathbb{R}^k$ by putting $F(D) = (q_1(D), \ldots, q_k(D))$. In this case $\Delta(F) \leqslant \sum_{i=1}^{k} \Delta(q_i) = k$.

Statistical Queries. Throughout this thesis we use the term *statistical query* to describe a *normalized* counting query. That is a statistical query $q\colon \mathscr{D} \to [0,1]$ counts the fraction of database items satisfying a predicate $P\colon \mathscr{U} \to [0,1]$. Note that here we allow the predicate to assume any real number between 0 and 1. Formally,

$$q(D) \stackrel{\text{def}}{=} \frac{1}{|D|} \sum_{u \in \mathscr{U}} P(u)D(u). \tag{2.3}$$

The motivation for this terminology stems from learning theory as we will see in Section 2.9.

Note that due to the difference in normalization statistical queries have sensitivity $1/n$ whereas counting queries have sensitivity 1.

Linear queries Linear queries are a generalization of counting and statistical queries. We use the term *linear query* to refer to a query that is a linear function of the histogram space. It would therefore be more accurate to say *linear histogram query*, but we prefer the shorter form. Specifically, a linear query is a linear mapping $f\colon \mathbb{R}^N \to \mathbb{R}$ defined on the N-dimensional histogram space. Since f is linear we can also think of it as a vector $f \in \mathbb{R}^N$ in which case we put $f(x) \stackrel{\text{def}}{=} \langle f, x \rangle$ where $\langle f, x \rangle$ denotes the usual inner product on $\mathbb{R}^N$. To control the sensitivity of the query we will assume that its coefficients are in $[0,1]$, i.e., $f\colon \mathbb{R}^N \to [0,1]$. In this case we have $\Delta(f) \leqslant 1$. The choice of $[0,1]$ is somewhat arbitrary. Any bounded interval would work for our purpose and the sensitivity would be the length of the interval. A multi-dimensional linear query is a linear mapping $f\colon \mathbb{R}^N \to \mathbb{R}^k$ and its sensitivity scales accordingly.

It should be clear at this point that counting queries and statistical queries are a special case of linear queries. Indeed, given a predicate $P\colon \mathscr{U} \to [0,1]$ we can define a corresponding linear query $f \in [0,1]^N$ by putting $f_u = P(u)$ for all $u \in [N]$. Furthermore, a linear query $f \in [0,1]^N$ on a normalized histogram x is nothing more than a statistical query on the database represented by the histogram.

2.3 Differentially private query release

Differentially private query release is the following problem: A trusted party called *database curator* is given a database $D \in \mathscr{D}$ of sensitive information and privacy parameters $\varepsilon > 0, \delta \geqslant 0$. An untrusted party called *data analyst*

specifies a sequence of queries $q_1, \ldots, q_k : \mathscr{D} \to \mathscr{R}$. The curator outputs answers $a_1, \ldots, a_k \in \mathscr{R}$ where $a_t \in \mathscr{R}$ is the curator's answer to query q_t. In general, the curator need not specify a list of answers but could rather provide a data structure encoding her answers. We will sometimes call such a data structure a *synopsis*.

The requirement is that the output of the curator satisfy (ε, δ)-differential privacy. Subject to this constraint the curator's goal is to maximize the *usefulness* of her answers. We will use varying notions of utility throughout this work.

We distinguish between *non-interactive* and *interactive* query release.

2.3.1 Non-interactive query release

In the non-interactive query release problem, all queries $q_1, \ldots, q_k : \mathscr{D} \to \mathscr{R}$ are given to curator up front. The curator chooses her answers with full knowledge of the entire query sequence. Note that we can always think of a sequence of queries in the non-interactive setting as a single query $q : \mathscr{D} \to \mathscr{R}^k$.

The right notion of accuracy often depends on the application and the setting of $\mathscr{R}$. When it comes to real-valued queries $q : \mathscr{D} \to \mathbb{R}^k$, a standard choice is the following.

Definition 2.3.1. (Accuracy) A mechanism $M : \mathscr{D} \to \mu(\mathbb{R}^k)$ is *α-accurate* in ℓ_p-norm on a query $q : \mathscr{D} \to \mathbb{R}^k$ if for every data set $D \in \mathscr{D}$ we have

$$\mathop{\mathbb{E}}_{v \sim M(D)} \|q(D) - v\|_p \leqslant \alpha \,. \tag{2.4}$$

We will sometimes refer to (2.4) as the *ℓ_p-accuracy* or *ℓ_p-error* of the mechanism M. If no norm is specified we take $p = \infty$.

When M outputs a synopsis rather than numerical answers we will measure accuracy with respect to the answers encoded by the synopsis.

We will encounter some simple variants of this definition throughout the thesis. For example, sometimes we will introduce another parameter β which quantifies the probability with which the algorithm fails to be α-accurate.

2.3.1.1 Synthetic data

An appealing variant of the non-interactive setting arises when we require that the output of the curator is itself a database $D^* \in \mathscr{D}$ encoding useful answers to the given queries. In this case we will call D^* *synthetic data*. Note that this is a strong but useful requirement on the output format of the algorithm. Indeed, synthetic data guarantees the compatibility of the

output with existing (non-private) tools for analyzing databases. Synthetic data also guarantees *consistency* which means that answers to related queries are not contradictory. For example, asking two counting queries defined by complementary predicates will result in two answers adding up to the number of participants. Such consistency cannot be expected in general from differentially private mechanisms that simply output noisy answers.

2.3.2 Interactive query release

In the *interactive* setting the curator and the analyst interact in k rounds. In round $t \in \{1,\ldots,k\}$, the curator chooses a query $q_t : \mathscr{D} \to \mathscr{R}$ from some class of queries $\mathcal{Q}$ and the curator provides an answer a_t. The query q_t may be chosen adaptively based on the previous interaction $q_1, a_1, \ldots, q_{t-1}, a_{t-1}$.

Note that all other things being equal, an interactive mechanism is preferable to a non-interactive one: if we have an interactive mechanism, even if the queries are all specified in advance, we can still run the interactive mechanism on the queries, one by one, and obtain privacy-preserving answers to all of them.

The interaction of the curator with a fixed data analyst specifies a non-interactive mechanism $M : \mathscr{D} \to \mathscr{R}^*$ where $\mathscr{R}^*$ is the set of all possible *transcripts*. A transcript is a sequence $(q_1, a_1, \ldots, q_k, a_k)$ where $q_i \in \mathcal{Q}$ and $a_i \in \mathscr{R}$. When we say that an interactive mechanism satisfies (ε, δ)-differential privacy we mean that for every data analyst the induced mechanism over transcripts satisfies (ε, δ)-differential privacy.

Similarly, the accuracy of an interactive mechanism is the worst-case taken over all adversaries of the accuracy of the resulting mechanism $M : \mathscr{D} \to \mathscr{R}^*$. A transcript specifies answers to the k given queries in the obvious way. Hence, any notion of accuracy in the non-interactive setting (such as Definition 2.3.1) gives rise to a corresponding notion in the interactive setting.

We will extend the formal discussion of the interactive setting in Chapter 4.

2.4 Laplace and composition

In this section we introduce two of the most basic and useful tools in differential privacy: the *Laplacian mechanism* and the *composition* theorem.

To introduce the Laplacian mechanism we need to define the Laplace distribution. We let $Lap(\sigma)$ denote the one-dimensional Laplacian distribution centered at 0 with scaling σ and corresponding density

$$f(x) = \frac{1}{2\sigma} \exp\left(-\frac{|x|}{\sigma}\right).$$

Note that $\mathbb{E}\, Lap(\sigma) = 0$.

Theorem 2.4.1. *Let $q\colon \mathscr{D} \to \mathbb{R}$ be a real-valued query of sensitivity 1. Then the mechanism $M\colon \mathscr{D} \to \mu(\mathbb{R})$ defined as $M(D) = q(D) + Lap(\Delta(q)/\varepsilon)$ satisfies $(\varepsilon, 0)$-differential privacy (where $\Delta(q)$ was defined in Definition 2.2.1).*

Proof. It suffices to compare the density f of $M(D)$ with the density g of $M(D')$ for any two neighboring D, D' at any point $x \in \mathbb{R}$. Without loss of generality we may assume that $q(D) = 0$. Put $c = q(D')$ and note that $|c| \leqslant \Delta = \Delta(q)$. Hence,

$$\frac{f(x)}{g(x)} = \frac{\exp(-\varepsilon|x|/\Delta)}{\exp(-\varepsilon|x-c|/\Delta)} \leqslant \frac{\exp(-\varepsilon|x|/\Delta)}{\exp(-\varepsilon|x|/\Delta)\exp(-\varepsilon|c|/\Delta)} \leqslant \exp(\varepsilon).$$

■

To analyze the error of the Laplacian mechanism we need the following simple lemma which is easy to verify.

Lemma 2.4.2. *The Laplace distribution satisfies:*

1. $\mathbb{E}\,|Lap(\sigma)| = \sigma$
2. $\mathbb{P}\{|Lap(\sigma)| \geqslant \tau\} = \exp(-\tau/\sigma)$.

Proof. Indeed,

$$\mathbb{E}\,|Lap(\sigma)| = 2\int_0^\infty x \cdot f(x)\,\mathrm{d}x = \sigma \int_0^\infty x \exp(-x)\,\mathrm{d}x = \sigma\,,$$
$$\mathbb{P}\{|Lap(\sigma)| \geqslant \tau\} = 2\int_\tau^\infty f(x)\,\mathrm{d}x = \frac{1}{\sigma}\int_\tau^\infty \exp(-x/\sigma)\,\mathrm{d}x = \exp(-\tau/\sigma).$$

■

Corollary 2.4.3. *There is an $(\varepsilon, 0)$-differentially private mechanism that is $O(\Delta/\varepsilon)$-accurate on every real-valued query $q\colon \mathscr{D} \to \mathbb{R}$ of sensitivity Δ.*

Theorem 2.4.1 extends straightforwardly to the multi-dimensional query case. Here we take an alternative route that goes through so-called *composition* theorems. Informally speaking, these theorems show that if we compose multiple differentially private mechanisms, the privacy parameters simply add up in the following sense.

Theorem 2.4.4. *Let $M_1\colon \mathscr{D} \to \mu(\mathscr{R})$ denote an $(\varepsilon_1, \delta_1)$-differentially private mechanism and let $M_2\colon \mathscr{D} \to \mu(\mathscr{R})$ denote an $(\varepsilon_2, \delta_2)$-differentially private mechanism. Then, the mechanism $M\colon \mathscr{D} \to \mu(\mathscr{R}\times\mathscr{R})$ defined as $M(D) = (M_1(D), M_2(D))$ satisfies $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-differential privacy.*

A stronger composition theorem was shown for (ε, δ)-differential privacy by [DRV].

Theorem 2.4.5 ([DRV]). *Let $\varepsilon_0 > 0, \delta > 0$ and $k \in \mathbb{N}$. Suppose that for each $i \in [k]$, we have that $M_i \colon \mathscr{D} \to \mu(\mathscr{R})$ is an (ε, δ)-differentially private mechanism. Then, for every $\delta' > 0$ the mechanism $M = (M_1, \ldots, M_k)$ is $(\varepsilon', k\delta + \delta')$-differentially private where*

$$\varepsilon' = \sqrt{2k \ln(1/\delta)}\varepsilon + k\varepsilon \cdot (e^{\varepsilon} - 1).$$

These composition theorems hold in a much more general and powerful composition framework that was formalized by Dwork et al. [DRV]. Intuitively speaking, the general framework allows the mechanism M_i to be selected adaptively and adversarially based on the outcomes of $M_1, \ldots, M_{i-1}$. We stated the theorem in the special case where composition simply means non-adaptive concatenation of k mechanisms. This special case already gives the following useful corollary.

Corollary 2.4.6. *Let $\varepsilon, \delta > 0$. For every sequence of k real-valued queries $q_1, \ldots, q_k \colon \mathscr{D} \to \mathbb{R}$ each of sensitivity Δ there is an*

1. *$(\varepsilon, 0)$-differentially private mechanism with $O(\Delta k \log k/\varepsilon)$- accuracy,*
2. *(ε, δ)-differentially private with $O\left(\Delta\sqrt{k \log(1/\delta)} \log k/\varepsilon\right)$-accuracy.*

Proof. To show the first part note that we can add noise $Lap(k\Delta/\varepsilon)$ to each of the queries. By Corollary 2.4.3 each instantiation is $(\varepsilon/k, 0)$-differentially private and $O(\Delta k/\varepsilon)$-accurate. Using Lemma 2.4.2, it is not hard to argue that the expected maximum error among all k queries is bounded by $O(\Delta k \log k/\varepsilon)$. Theorem 2.4.4 then concludes the first part of the proof.

For the second part we add noise $Lap(\sqrt{k}\Delta/c\varepsilon)$ for sufficiently small constant $c > 0$. Now each instantiation of is $(c\varepsilon/\sqrt{k}, 0)$-differentially private. Applying Theorem 2.4.5 with sufficiently small c it follows that the composition is (ε, δ)-differentially private. Accuracy is argued as before. ∎

Remark 2.4.7. When we refer to the *Laplacian mechanism* from here on we will usually mean the algorithm as described in the first part of Corollary 2.4.6. The second part of Corollary 2.4.6 can be shown more directly using the *Gaussian mechanism* [DMNS, DKM+]. The Gaussian mechanism adds suitably scaled noise chosen from a Gaussian (rather than Laplacian) random variable to the answers. We omit a formal statement and analysis here.

2.5 The exponential mechanism

Another basic tool is the exponential mechanism of McSherry and Talwar [MT]. The exponential mechanism gives a differentially private general purpose method to select an object from a domain $\mathscr{R}$ according to some score function $s\colon \mathscr{D} \times \mathscr{R} \to \mathbb{R}$ and a base measure ν on $\mathscr{R}$.

Definition 2.5.1. Formally, we define the *exponential mechanism* $E_s^\varepsilon\colon \mathscr{D} \to \mu(\mathscr{R})$ by putting $E_s^\varepsilon(D)$ to be the measure given by the density function

$$f(r) = Z^{-1} \exp(\varepsilon s(D,r))\nu(r), \tag{2.5}$$

where $Z = \int \exp(\varepsilon s(D,r))\nu(r)$. Here, we require that the integral defining the normalization constant Z exists. We let $\Delta(s) = \sup_{r,D,D'} |s(D,r) - s(D',r)|$ where the supremum runs over all neighboring databases $D, D' \in \mathscr{D}$ and all $r \in \mathscr{R}$.

When $\mathscr{R}$ is a finite domain we will usually take ν to be the uniform counting measure, i.e., $\nu(S) = |S|/|\mathscr{R}|$ for all $S \subseteq \mathscr{R}$.

The next lemma quantifies the privacy and utility guarantee achieved by the exponential mechanism.

Lemma 2.5.2 ([MT])**.** *The exponential mechanism E_s^ε satisfies $(\varepsilon\Delta(s), 0)$-differential privacy. Furthermore, for every D with $\mathrm{OPT} = \sup_{r\in\mathscr{R}} s(D,r)$ and $P = E_s^\varepsilon(D)$ we have*

$$\mathop{\mathbb{P}}_{r\sim P}\{s(D,r) \leqslant \mathrm{OPT} - 2t\} \leqslant \frac{\exp(-\varepsilon t)}{\nu(S_t)}, \tag{2.6}$$

where $S_t = \{r\colon s(D,r) > \mathrm{OPT} - t\}$.

A useful implication of the previous lemma is that when ν is the uniform counting measure over a finite domain $\mathscr{R}$, then $\nu(S_t) \geqslant 1/|\mathscr{R}|$ and

$$\mathop{\mathbb{E}}_{r\sim P} s(D,r) \leqslant \mathrm{OPT} - O\left(\frac{\log|\mathscr{R}|}{\varepsilon}\right).$$

We will use the exponential mechanism and the previous lemma in particular in Chapter 5.

2.6 Mechanisms for massive query sets

The Laplacian mechanism as described in Corollary 2.4.6 ceases to give a meaningful accuracy guarantee as the number of queries grows well above the database size. Can we give answers to $k \gg n$ queries with non-trivial accuracy? This question was answered in the affirmative by Blum, Ligett and Roth [BLR].

Theorem 2.6.1 ([BLR]). *There is an $(\varepsilon, 0)$-differentially private mechanism that answers any given set of k counting queries on a database of size n over a universe of size N with accuracy*

$$O\left(\frac{n^{2/3}\log^{1/3} k \log^{1/3} N}{\varepsilon^{1/3}}\right). \tag{2.7}$$

The key observation behind this theorem is that $\mathscr{D}_n$, i.e., the set of databases of size n, is well approximated by a much smaller set $\mathscr{D}^* \subseteq \mathscr{D}_n$. Specifically, for every set of k counting queries there is a set $\mathscr{D}^*$ of size $|\mathscr{D}^*| \leqslant \exp(O(\alpha^{-2}\log k \log N))$ such that for every $D \in \mathscr{D}_n$ there is $D' \in \mathscr{D}^*$ such that D and D' agree on the k queries up to a maximum error of αn. The algorithm of [BLR] then outputs a synthetic data set $D^* \in \mathscr{D}^*$ chosen from the exponential mechanism with a score function proportional to the negative of the maximum error of D^* on the k queries. By Lemma 2.5.2, the database D^* has expected maximum error $O(\alpha n + \alpha^{-2}\log k \log N)$. Optimizing α gives the bound in (2.7).

Unfortunately, the algorithm described above is highly inefficient. The exponential mechanism requires time $N^{\omega(1)}$ as soon as $\alpha = o(1)$. Moreover the mechanism is non-interactive. It is natural to ask if a similar result can be achieved in the interactive query release setting. Each of these aspects has since been the subject of subsequent works. Dwork et al. [DNR+] improved the running time to polynomial in N and k with an accuracy of $O(\sqrt{n}k^{o(1)})$ in the non-interactive setting. Dwork, Rothblum and Vadhan [DRV] then improved the dependence on the error in terms of n and k to $O(\sqrt{n}\cdot \operatorname{polylog} k)$. Moreover, the result holds for arbitrary low-sensitivity queries. Both results relax the privacy guarantee to (ε, δ)-differential privacy.

Roth and Roughgarden [RR] demonstrated that surprisingly also in the *interactive* setting there is an (ε, δ)-differentially private mechanism with error $O(n^{2/3}\operatorname{polylog} k)$. However, their mechanism does not improve over the algorithm of Blum, Ligett and Roth in terms of running time.

In Chapter 4, we will see that one can simultaneously remedy both shortcomings and obtain an interactive mechanism with running time linear in N on each query and error $O(\sqrt{n\log k})$. We also give an interactive mechanism achieving $(\varepsilon, 0)$-differential privacy with the error bound stated in Theorem 2.6.1.

The error $O(\sqrt{n\log k})$ matches what is known as the *statistical sampling error*. The statistical sampling error arises free of any privacy concerns when we sample a data set D of n data items from a distribution X over the universe $\mathscr{U}$. Indeed, the expected deviation of a single insensitive query $q\colon \mathscr{D} \to \mathbb{R}$ from its expected value is $O(\sqrt{n})$. The expected maximum deviation of k insensitive queries is similarly $O(\sqrt{n\log k})$. We will see in Section 2.8 that in fact

$O(\sqrt{n \log k})$ is also a lower bound that any differentially private mechanism must obey.

A priori one might hope to further improve the runtime from poly(N) to poly(n). In Section 2.8 we will discuss known obstacles to making such progress.

2.7 Results for other classes of queries

Some specific classes of queries can be answered with smaller error. Nissim, Raskhodnikova and Smith [NRS] show that one can add noise proportional to a smoothed version of the *local sensitivity* of the query, which can be much smaller than the ℓ_1-sensitivity (Definition 2.2.1) for some *non-linear* queries.

There has been a considerable amount of work on other classes of queries and release problems. We mention some of them here. For example, Feldman et al. [FFKN] consider the problem of constructing differentially private geometric data structures known as *core sets*. Specifically, they construct private core sets for the k-median problem, enabling approximate computation of the k-median cost of any set of k facilities in $\mathbb{R}^d$. Gupta et al. [GLM+] initiated a study of differentially private combinatorial optimization. Differentially private search logs were studied in [KKMN, GMW+] motivated by the AOL search log debacle [BZ]. Mironov and McSherry considered differentially private recommender systems [MM] motivated by the Netflix problem described in the introduction.

A particularly well-studied class of queries are *contingency tables* which we will discuss next.

2.7.1 Contingency tables and conjunctions

Contingency tables (aka *marginal tables* or *data cubes*) are a notable special case of counting queries. Consider the universe $\mathscr{U} = \{0,1\}^d$. That is, each data item is a bit string consisting of d binary attributes. A contingency table of width w corresponding to a subset $W \subseteq [d]$ of size $|W| = w$ is a collection of 2^w counting queries. Each of these 2^w counting query is *conjunction query* corresponding to a subset $S \subseteq W$ and counts how many items in the data base have the bits in S set to 1 and the bits in $W \setminus S$ set to 0.

Conjunctions themselves are an interesting class of counting queries. Even monotone conjunctions allow us to learn *covariances* between attributes in the data set. This is an important statistical tool.

Barak et al. [BCD+] gave an algorithm for producing differentially private synthetic data for width-w contingency tables (thus ensuring consistency of all answers). However, the error scaled with $d^{O(w)}$ and the runtime was

polynomial in 2^d. Fienberg et al. [FRY] evaluated the techniques of Barak et al. on real-world data sets and concluded that differential privacy was not (yet) suitable for releasing contingency tables in a practical way. In Chapter 5 we will revisit this claim and see that indeed the multiplicative weights framework from Chapter 4 does yield a practical release mechanism for contingency tables on the data sets studied by Fienberg et al. Other practical mechanisms were studied in [DWHL].

In Chapter 6 we will see a release mechanism with runtime poly(d) that achieves a non-trivial accuracy guarantee (yet no consistency). The set of all w-way conjunctions can also be released privately using the Laplacian mechanism which results in an error of $O(d^{w/2})$ when shooting for (ε, δ)-differential privacy. This shows that for subconstant accuracy the database size n has to be $\omega(d^{w/2})$. In Chapter 7 we show that subconstant accuracy can be achieved already when $n \gg d^{\widetilde{O}(\sqrt{w})}$.

2.8 Lower bounds and hardness results

The works of Dinur and Nissim [DN1] and Dwork and Nissim [DN2] initiated the study of lower bounds on the amount of noise mechanisms must add in order to protect against *blatant non-privacy*. Informally, blatant non-privacy describes a privacy breach in which an adversary is able to reconstruct a significant part of the database from the query answers released by the mechanism. Dinur and Nissim consider the setting where the database is a vector $y \in \{0,1\}^n$ containing a single private bit for each individual. The queries asked by the analyst are subset sum queries counting how many 1's there are in a subset of the individuals. Dinur and Nissim show that any mechanism adding noise $o(\sqrt{n})$ to $\widetilde{O}(n)$ queries fails to protect against blatant non-privacy. This implies that as the data curator answers more and more questions, the amount of error needed per answer must grow to provide any kind of privacy guarantee.

To give a flavor for how one would prove such a result, think of n random subset sum queries as specifying a random binary $n \times n$ matrix A. Indeed a subset sum query is just a vector $f \in \{0,1\}^n$. The answer of a subset sum query on a database $y \in \{0,1\}^n$ is just the inner product $\langle f, y \rangle$. Now, suppose a database curator unaware of [DN1] perturbs the answers Ay with a noise vector $e \in \mathbb{R}^n$ such that every coordinate of e is bounded by $o(\sqrt{n})$ in magnitude. An adversary can now observe the resulting vector $z = Ay + e$, and efficiently compute the vector $y' = A^{-1}z$. Note that A is invertible with overwhelming probability and moreover, by a standard fact from random matrix theory, its smallest eigenvalue is $\Omega(\sqrt{n})$. Hence, the spectral norm of A^{-1} satisfies

$\|A^{-1}\|_2 \leqslant O(1/\sqrt{n})$ and thus

$$\|y' - y\|_2 = \|A^{-1}e\|_2 \leqslant \|A^{-1}\|_2 \cdot \|e\|_2 = O\left(\frac{\|e\|_2}{\sqrt{n}}\right) = o\left(\sqrt{n}\right).$$

In the last step we used that $\|e\|_2 = o(n)$ by our assumption. Since almost all vectors $y \in \{0,1\}^n$ have norm $\|y\|_2 = \Omega(\sqrt{n})$, this can be thought of as a non-trivial *reconstruction attack* on the database y.

The kind of trade-off shown by Dinur and Nissim was strengthened and simplified in subsequent work [DN2, DMT, DY]. In our setting where the data set is an element of $\mathscr{D}$ and the curator is allowed to ask k (random) counting queries, it can be shown that noise $\Omega(\min\{\sqrt{k}, \sqrt{n \log k}\})$ is necessary to prevent blatant non-privacy. Recall that $O(\sqrt{n \log k})$ is also the statistical sampling error. Viewed in this light these results do not show that ensuring privacy requires paying a prize beyond the sampling error.

Kasiviswanathan, Rudelson, Smith and Ullman [KRSU] show lower bounds of similar nature for contingency tables. Specifically, they show that noise $\widetilde{\Omega}(\min\left\{\sqrt{n}, d^{w/2}\right\})$ is necessary for privately releasing all width w contingency tables over $\mathscr{U} = \{0,1\}^d$. Their lower bound also holds for (ε, δ)-differential privacy and is tight when ε and δ are constants. However, there is no dependence on $1/\varepsilon$ or $1/\delta$ in the bound. We remark that their proof follows the approach outlined above. Specifically, they analyze the smallest singular value of certain matrices arising from contingency tables.

For the case of a single insensitive query $q \colon \mathscr{D} \to \mathbb{R}$, Ghosh, Roughgarden and Sundararajan [GRS] show that adding Laplace noise is in fact optimal in a general decision-theoretic framework. However, such a result (even for mechanisms other than Laplace) is impossible for multiple queries [BN].

An important question left open by all these works is whether there exist stronger lower bounds specifically for differential privacy. In Chapter 3 we will answer this question in the affirmative. We remark that using the techniques presented in Chapter 3, De [De] was able to further improve upon the lower bounds discussed here in several aspects.

A lower bound specific to $(\varepsilon, 0)$-differential privacy was shown independently of our work by Beimel, Kasiviswanathan and Nissim [BKN]. They show a lower bound of $\Omega(\log k/\varepsilon)$ on the accuracy of any $(\varepsilon, 0)$-differentially private mechanism for a specific set of k counting queries over a universe of size $N = k$. Their proof uses an idea similar to our proof in Chapter 3. In Chapter 4 we will give a lower bound of the form $\Omega(\sqrt{n \log k \log N}/\varepsilon)$. This strengthens the result of Beimel et al. by a $\sqrt{n}$-factor.

2.8.1 Hardness of synthetic data

All of the algorithms mentioned in Section 2.6 have the appealing feature that they (can be used to) produce *synthetic data*. Recall, this means the algorithm actually produces a differentially private data set $D^* \in \mathscr{D}$ which encodes accurate answers to the queries.

Unfortunately, Dwork et al. [DNR+] showed that in general we cannot hope for differentially private *synthetic data* release in time polynomial in the size of the data set or even sublinear in the data universe. Their hardness results are based on a connection between creating synthetic data and so-called traitor tracing schemes from cryptography. In particular, the result mentioned is based on plausible cryptographic hardness assumptions unlike the error lower bounds mentioned above. The result of Dwork et al. employs a rather contrived set of queries. It is then natural to try to side-step this hardness result by considering restricted query classes. But a recent work of Ullman and Vadhan [UV] shows hardness even for very simple and natural query classes such as two-way conjunctions.

Nevertheless these hardness results only apply for synthetic data and, in fact, in Chapter 6 and Chapter 7 we will see avenues for side-stepping the *synthetic data barrier*.

2.9 Learning theory and differential privacy

Machine learning and privacy-preserving data analysis go hand in hand. Indeed, machine learning captures many of the computations that an analyst would like to perform on a data set. Informally, a learning algorithm has (limited) access to examples drawn from a distribution X over a universe $\mathcal{U}$. The examples are labeled according to some *concept* $c\colon \mathcal{U} \to \{0,1\}$ from a *concept class* $\mathcal{C} \subseteq \mathcal{U} \to \{0,1\}$. The goal of the learner is to find a *hypothesis* $h\colon \mathcal{U} \to \{0,1\}$ which agrees with c on almost the entire universe.

Blum et al. [BDMN] show that many learning algorithms reduce to the task of asking a number of counting queries on the data set. Hence, these algorithms fall directly into the model studied in this thesis. Most notably, they show that any learning algorithm operating in Kearns' statistical query (SQ) model [Kea] can be implemented from noisy counting queries. We will formally introduce the SQ model in the next section and reprove this result. There are few examples of algorithms that work in the general PAC model [Val], but do not have an SQ analog.

A principled study of what can be learned privately was initiated by Kasiviswanathan et al. [KLN+]. In particular, they show that ignoring computational constraints "anything" learnable is also privately learnable from

few samples. There has been much work on privacy-preserving learning algorithms, e.g., SVM [RBHT], logistic regression [CM], sample complexity for infinite concept classes [CH]

2.9.1 The statistical query model

In Kearns' the statistical query (SQ) model [Kea] an algorithm $\mathscr{A}^{\mathcal{O}}$ can access a distribution X over a universe $\mathscr{U}$ only through *statistical queries* to an oracle $\mathcal{O}$. That is, the algorithm may ask any query $q\colon \mathscr{U} \to [0,1]$ and the oracle may respond with any answer a satisfying $|a - \mathbb{E}_{u\sim X}\, q(u)| \leqslant \tau$. Here, τ is a parameter called the *tolerance* of the query.

In the context of differential privacy, the distribution X will typically be the uniform distribution over a data set D of size n. A statistical query in the SQ model is then just the same as a statistical query on a database as defined in Section 2.2.1.

The original motivation behind SQ model is that algorithms designed in this model are more noise tolerant than traditional PAC learning algorithms. This is also the reason why it is not difficult to turn SQ algorithms into differentially private algorithms using a suitable oracle. This observation is has been used previously, for example by Blum et al. [BDMN] and Kasiviswanathan et al. [KLN+].

Proposition 2.9.1. *Let $\mathscr{A}$ denote an algorithm that requires k statistical queries of tolerance τ. Let $\mathcal{O}$ denote the oracle that outputs $q(D) + Lap(k/n\varepsilon)$ for some $D \in \mathscr{D}$. Then, the algorithm $\mathscr{A}^{\mathcal{O}}$ satisfies $(\varepsilon, 0)$-differential privacy and with probability at least $1-\beta$, the oracle answers all k queries with error at most τ provided that*

$$|D| \geqslant \frac{k(\log k + \log(1/\beta))}{\varepsilon\tau}. \tag{2.8}$$

Proof. The first claim follows directly from Corollary 2.4.6. The second claim follows from concentration properties of the Laplace distribution. Indeed, by Lemma 2.4.2 and Equation 2.8, a single oracle answer violates the tolerance requirement with probability at most β/k The claim now follows by taking a union bound over all k queries. ∎

In other words, provided that the database is large enough, the learning algorithm $\mathscr{A}$ will continue to work as intended while also satisfying differential privacy.

Remark 2.9.2 (Sample complexity versus accuracy)**.** We saw in Proposition 2.9.1 that it is sometimes more convenient to quantify the guarantee of an algorithm in terms of a lower bound on the database size. This corresponds to a bound on the *sample complexity* of an algorithm as is typical in learning theory.

When it comes to privacy-preserving release mechanisms for statistical queries we generally have the choice between the two statements:

1. The algorithm is $\alpha(n)$-accurate where $\alpha(n)$ is a function tending to zero as $n \to \infty$.

2. The algorithm is α-accurate provided that $n \geqslant n_0(\alpha)$ where n_0 is a function tending to infinity as $\alpha \to 0$.

In Chapter 4 and Chapter 5 it will be more convenient to make statements of the first kind, while in Chapter 6 and Chapter 7 the latter will be more convenient.

2.10 Tools from probability theory

Let $x, y \in \mathbb{R}^N_+$ be two vectors with non-negative entries. We define the *relative entropy* or *Kullback-Leibler divergence* between x and y as:

$$\mathrm{RE}(x\|y) = \sum_{i \in [N]} x_i \log\left(\frac{x_i}{y_i}\right) + y_i - x_i \,. \tag{2.9}$$

This reduces to the more familiar expression $\sum_i x_i \log(\frac{x_i}{y_i})$ when $\sum_i x_i = \sum_i y_i = 1$ (in particular this happens when x, y correspond to distributions over $[N]$).

The following fact about relative entropy is well-known and easy to verify.

Fact 2.10.1. *For every $x, y \in \mathbb{R}^N_+$, we have $\mathrm{RE}(x\|y) \geqslant 0$. Equality holds if and only if $x = y$.*

We utilize the following lemma about the convexity of the KL-divergence.

Lemma 2.10.2. *Let P, Q be arbitrary distributions over a common probability space. Suppose there are distributions P_1, P_2, Q_1, Q_2 and $\lambda \in [0,1]$, so that $P = \lambda P_1 + (1-\lambda)P_2$ and $Q = \lambda Q_1 + (1-\lambda)Q_2$. Then,*

$$\mathop{\mathbb{E}}_{v \sim P} \log\left(\frac{P(v)}{Q(v)}\right) \leqslant \lambda \mathop{\mathbb{E}}_{v \sim P_1} \log\left(\frac{P_1(v)}{Q_1(v)}\right) + (1-\lambda) \mathop{\mathbb{E}}_{v \sim P_2} \log\left(\frac{P_2(v)}{Q_2(v)}\right). \tag{2.10}$$

In other words, $\mathrm{RE}(P\|Q) \leqslant \lambda \mathrm{RE}(P_1, Q_1) + (1-\lambda)\mathrm{RE}(P_2, Q_2)$.

The next lemma shows that $(\varepsilon, 0)$-differential privacy translates into relative entropy $2\varepsilon^2$. This lemma was shown in [DRV].

Lemma 2.10.3 ([DRV]). *Let P, Q be any two distributions on a common support $\mathscr{R}$ with density functions $\mathrm{d}P$ and $\mathrm{d}Q$, respectively. Suppose that*

$$\sup_{v \in \mathscr{R}} \log\left(\frac{P(v)}{Q(v)}\right) \leqslant \varepsilon .$$

Then,

$$\mathop{\mathbb{E}}_{v \sim P} \log\left(\frac{P(v)}{Q(v)}\right) \leqslant 2\varepsilon_0 .$$

We also use the following general large deviation bound.

Lemma 2.10.4 (Method of Bounded Differences). *Let $X_1, \ldots, X_m$ be an arbitrary set of random variables and let f be a function satisfying the property that for every $j \in [m]$ there is a number $c_j \geqslant 0$ such that*

$$\left|\mathbb{E}[f \mid X_1, X_2 \ldots, X_j] - \mathbb{E}[f \mid X_1, X_2, \ldots, X_{j-1}]\right| \leqslant c_j .$$

Then,

$$\mathbb{P}\{f > \mathbb{E} f + \lambda\} \leqslant \exp\left(-\frac{\lambda^2}{2\sum_{j \in [m]} c_j^2}\right). \tag{2.11}$$

2.10.1 Gamma Distribution

The *Gamma distribution* with shape parameter $k > 0$ and scale $\theta > 0$, denoted $\mathrm{Gamma}(k, \theta)$, is given by the probability density function

$$f(r; k, \theta) = r^{k-1} \frac{e^{-r/\theta}}{\Gamma(k)\theta^k} .$$

Here, $\Gamma(k) = \int e^{-r} r^{k-1} \, \mathrm{d}r$ denotes the Gamma function. We will need an expression for the moments of the Gamma distribution.

Fact 2.10.5. *Let $r \sim \mathrm{Gamma}(k, \theta)$. Then,*

$$\mathbb{E}[r^m] = \frac{\theta^m \Gamma(k+m)}{\Gamma(k)} . \tag{2.12}$$

Proof.

$$\begin{aligned} \mathbb{E}[r^m] = \int_{\mathbb{R}} r^{k+m-1} \frac{e^{-r/\theta}}{\Gamma(k)\theta^k} \, \mathrm{d}r &= \frac{1}{\Gamma(k)\theta^k} \int_{\mathbb{R}} (\theta r)^{k+m-1} e^{-r} \, \mathrm{d}\theta r \\ &= \frac{\Gamma(k+m)\theta^{k+m}}{\Gamma(k)\theta^k} = \frac{\Gamma(k+m)\theta^m}{\Gamma(k)} \end{aligned}$$

■

Chapter 3

On the Geometry of Differential Privacy

In this chapter we exhibit a connection between differential privacy and convex geometry. This connection leads to a nearly optimal trade-off between accuracy and privacy in the following general setting. Specifically, we will work with databases represented as fractional histograms in $\mathbb{R}^N$. We saw in Section 2.1.1 how every database has a natural representation as a histogram. We consider answering d linear queries (see Section 2.2). Throughout this chapter we denote the number of queries by d rather than k used in other chapters.

We can represent d linear queries as a linear mapping $F\colon \mathbb{R}^N \to \mathbb{R}^d$. We will restrict ourselves to linear maps F with coefficients in the interval $[-1,1]$. Thus we can represent F as a $d \times N$ matrix with entries in $[-1,1]$. In this work, we assume throughout that $d \leqslant N/2$. This is without loss of generality as we may always add zero coordinates to x. A mechanism is a mapping $M\colon \mathbb{R}^N \to \mu(\mathbb{R}^d)$. Recall that differential privacy places a constraint on the output distribution of M on any two databases $x, x' \in \mathbb{R}^N$ such that $\|x - x'\|_1 \leqslant 1$ and it asks that $\mathbb{P}\{M(x) \in S\}/\mathbb{P}\{M(x') \in S\} \leqslant \exp(\varepsilon)$ for every $S \subseteq \mathbb{R}^d$. We will analyze the ℓ_2-error of differentially private mechanisms as defined in Definition 2.3.1.

Note that here we require the mechanism to ignore the integrality of its input. That is, the mechanism must be defined on all points in $\mathbb{R}^N$ even those that do not correspond to databases. This is a stronger requirement on the mechanism as is standard. Hence, it only makes our upper bounds stronger. Our upper bound holds for any linear query on the histogram. As explained in Section 2.2, this includes some of the most well-studied and natural classes of queries in statistical data analysis.

For the lower bounds, this strengthening allows us to ignore the discretization issues that would arise in the usual definition. Building on the techniques presented in this chapter, De [De] recently showed that our lower bounds can be extended to hold for the usual definition.

3.1 Main results

We relate the accuracy of differentially private mechanisms to some geometric properties of the image of the unit ℓ_1-ball, denoted B_1^N, when applying the linear mapping F. We will denote the resulting convex polytope by $K = FB_1^N$. Our first result lower bounds the noise any ε-differentially private mechanism must add in terms of the *volume radius* of K, denoted $\mathrm{vr}(K)$. Here,

$$\mathrm{vr}(K) \stackrel{\mathrm{def}}{=} \left(\frac{\mathrm{Vol}(K)}{\mathrm{Vol}(B_2^d)}\right)^{1/d},$$

where B_2^d denotes the d-dimensional Euclidean ball.[1]

Theorem 3.1.1. *Let $\varepsilon > 0$ and suppose $F\colon \mathbb{R}^N \to \mathbb{R}^d$ is a linear map. Then, every ε-private mechanism M has error at least*

$$\Omega\left(\frac{d \cdot \mathrm{vr}(K)}{\varepsilon}\right), \tag{3.1}$$

where $K = FB_1^N$.

Recall, the term *error* refers to the expected Euclidean distance between the output of the mechanism and the correct answer to the query F.

We then describe a differentially private mechanism whose error depends on the expected ℓ_2-norm of a randomly chosen point in K. Our mechanism is an instantiation of the exponential mechanism [MT] with the score function defined by the (negative of the) norm $\|\cdot\|_K$, that is the norm which has K as its unit ball. Hence, we will refer to this mechanism as the K-norm mechanism. Note that as the definition of this norm depends on the query F, so does the output of our mechanism. The error of this mechanism can be described in terms of the *mean radius* of K defined as

$$\mathrm{mr}(K) \stackrel{\mathrm{def}}{=} \mathop{\mathbb{E}}_{z \in K} \|z\|_2$$

where z is drawn uniformly at random from K.

Theorem 3.1.2. *Let $\varepsilon > 0$ and suppose $F\colon \mathbb{R}^N \to \mathbb{R}^d$ is a linear map with $K = FB_1^N$. Then, the K-norm mechanism is ε-differentially private and has error at most*

$$O\left(\frac{d \cdot \mathrm{mr}(K)}{\varepsilon}\right). \tag{3.2}$$

[1] Volume radius is typically defined as $\mathrm{Vol}(K)^{1/d}$. The different normalization we chose will be convenient for us.

As it turns out, when F is a random Bernoulli ± 1 matrix our upper bound matches the lower bound up to constant factors. In this case, K is a random polytope and its volume and mean radius have been determined rather recently. Specifically, we apply a volume lower bound of Litvak et al. [LPRTJ], and an upper bound on the mean radius due to Klartag and Kozma [KK]. Quantitatively, we obtain the following theorem which gives tight upper and lower bounds.

Theorem 3.1.3. *Let $\varepsilon > 0$ and $d \leqslant N/2$. Then, for almost all matrices $F \in \{-1,1\}^{d\times N}$,*

1. *any ε-differentially private mechanism M has error $\Omega\left(\frac{d}{\varepsilon} \cdot \min\left\{\sqrt{d}, \sqrt{\log(N/d)}\right\}\right)$.*
2. *the K-norm mechanism is ε-differentially private with error $O\left(\frac{d}{\varepsilon} \cdot \min\left\{\sqrt{d}, \sqrt{\log(N/d)}\right\}\right)$.*

We remark that Litvak et al. [LPRTJ] also give an explicit construction of a mapping F realizing the lower bound.

More generally, we can relate our upper and lower bounds whenever the body K is in *approximately isotropic position*. Informally, this condition implies that $\mathrm{mr}(K) \sim \mathrm{vr}(K)L_K$ where L_K denotes the so-called *isotropic constant* which is defined in Section 3.5.

Theorem 3.1.4. *Let $\varepsilon > 0$ and suppose $F\colon \mathbb{R}^N \to \mathbb{R}^d$ is a linear map such that $K = FB_1^N$ is in approximately isotropic position. Then, the K-norm mechanism is ε-differentially private with error at most $O(\varepsilon^{-1}\mathrm{vr}(dK))$.*

Notice that the bound in the previous theorem differs from the lower bound by a factor of L_K. A central conjecture in convex geometry, sometimes referred to as the "Hyperplane Conjecture" or "Slicing Conjecture" states that $L_K = O(1)$. See, e.g., [MP, Gia, KK] for further information on the subject.

Unfortunately, in general the polytope K could be very far from isotropic. In this case, both our volume-based lower bound and the K-norm mechanism can be quite far from optimal. We give a recursive variant of our mechanism and a natural generalization of our volume-based lower bound which are nearly optimal even if K is non-isotropic.

Theorem 3.1.5. *Let $\varepsilon > 0$. Suppose $F\colon \mathbb{R}^N \to \mathbb{R}^d$ is a linear map. Further, assume the Hyperplane Conjecture. Then, the mechanism introduced in Section 3.6 is ε-differentially private and has error at most $O(\log^{3/2} d) \cdot \mathrm{VolLB}(K,\varepsilon)$. where $\mathrm{VolLB}(K,\varepsilon)$ is a lower bound on the error of the optimal ε-differentially private mechanism.*

While we restricted our theorems to $F \in [-1,1]^{d\times N}$, they apply more generally to any linear mapping F.

Mechanism	ℓ_2**-error**	**privacy**	reference
Laplacian noise	$\varepsilon^{-1} d\sqrt{d}$	ε	[DMNS]
K-norm	$\varepsilon^{-1} d\sqrt{\log(N/d)}$	ε	here
lower bound	$\Omega(\varepsilon^{-1} d)$	(ε, δ)	[DN1]
lower bound	$\Omega(\varepsilon^{-1} d)\min\left\{\sqrt{\log(N/d)}, \sqrt{d}\right\}$	ε	here

Figure 3.1: Summary of results in comparison to best previous work for d *random* linear queries each of sensitivity 1 where $1 \leqslant d \leqslant n$. Note that informally the average per-coordinate error is smaller than the stated bounds by a factor of $\sqrt{d}$. Here, (ε, δ)-differential privacy refers to a weaker approximate notion of pricacy introduced later. Our lower bound does not apply to this notion.

Efficient Mechanisms. Our mechanism is an instantiation of the exponential mechanism and involves sampling random points from rather general high dimensional convex bodies. This is why our mechanism is not efficient as it is. However, we can use rapidly mixing geometric random walks for the sampling step. These random walks turn out to approach the uniform distribution in a metric that is strong enough for our purposes. It will follow that both of our mechanisms can be implemented in polynomial time.

Theorem 3.1.6. *The mechanisms given in Theorem 3.1.2 and Theorem 3.1.5 can be implemented in time polynomial in $n, 1/\varepsilon$ such that the stated error bound remains the same up to constant factors, and the mechanism achieves ε-differential privacy.*

We note that our lower bound VolLB can also be approximated up to a constant factor. Together these results give polynomial time computable upper and lower bounds on the error of any differentially private mechanism, that are always within an $O(\log^{3/2} d)$ of each other.

Figure 3.1 summarizes our results. Note that we state our bounds in terms of the total ℓ_2-error, which informally is a $\sqrt{d}$ factor larger than the average per-coordinate error.

Related work. For an extensive discussion of previous lower bounds see Section 2.8.

We add that our lower bounds (explained in Section 3.2) are in short based on a *packing argument*. This kind of argument has since found several further applications in proving lower bounds in differential privacy. Subsequent to our work, De [De] showed that our lower bounds also hold under the weaker requirement that the mechanism is only defined on non-negative integer points $\mathbb{Z}_+^N$ rather than $\mathbb{R}^N$.

We remark that an idea similar to the packing argument was also used independently of our work in a lower bound by Beimel, Kasiviswanathan and Nissim [BKN].

3.1.1 Overview and organization of this chapter

We prove our lower bound in Section 3.2. Given a query $F\colon \mathbb{R}^N \to \mathbb{R}^d$, our lower bound depends on the d-dimensional volume of $K = FB_1^N$. If the volume of K is large, then a packing argument shows that we can pack exponentially many points inside K so that each pair of points is far from each other. We then scale up K by a suitable factor λ. By linearity, all points within λK have preimages under F that are still λ-close in ℓ_1-distance. Hence, the definition of ε-differential privacy (by transitivity) enforces some constraint between these preimages. We can combine these observations so as to show that any differentially private mechanism M will have to put significant probability mass in exponentially many disjoint balls. This forces the mechanism to have large expected error.

We then introduce the K-norm mechanism in Section 3.3. Our mechanism computes Fx and then adds a noise vector to Fx. The key point here is that the noise vector is not independent of F as in previous works. Instead, informally speaking, the noise is tailored to the exact shape of $K = FB_1^N$. This is accomplished by picking a particular noise vector v with probability proportional to $\exp(-\varepsilon\|Fx - v\|_K)$. Here, $\|\cdot\|_K$ denotes the (Minkowski) norm defined by K. While our mechanism depends upon the query F, it does *not* depend on the particular database x. We can analyze our mechanism in terms of the expected Euclidean distance from the origin of a random point in K, i.e., $\mathbb{E}_{z\in K}\|z\|_2 = \mathrm{mr}(K)$. Arguing optimality of our mechanism hence boils down to relating $\mathrm{mr}(K)$ to the volume of K.

Indeed, using several results from convex geometry, we observe that our lower and upper bounds match up to constant factors when F is drawn at random from $\{-1,1\}^{d\times N}$. As it turns out the polytope K can be interpreted as the symmetric convex hull of the row vectors of F. When F is a random matrix, K is a well-studied random polytope. Some recent results on random polytopes give us suitable lower bounds on the volume and upper bounds on the average Euclidean norm. More generally, our bounds are tight whenever K is in isotropic position (as pointed out in Section 3.5). This condition intuitively gives a relation between volume and average distance from the origin. Our bounds are actually only tight up to a factor of L_K, the isotropic constant of K. A well-known conjecture from convex geometry, known as the Hyperplane Conjecture or Slicing Conjecture, implies that $L_K = O(1)$.

The problem is that when F is not drawn at random, K could be very far

from isotropic. In this case, the K-norm mechanism by itself might actually perform poorly. We thus give a recursive variant of the K-norm mechanism in Section 3.6 which can handle non-isotropic bodies. Our approach is based on analyzing the covariance matrix of K in order to partition K into parts on which our earlier mechanism performs well. Assuming the Hyperplane conjecture, we derive bounds on the error of our mechanism that are optimal to within polylogarithmic factors.

The costly step in both of our mechanisms is sampling uniformly from high dimensional convex bodies such as $K = FB_1^N$. To implement the sampling step efficiently, we will use geometric random walks. It can be shown that these random walks approach the uniform distribution over K in polynomial time. We will actually need convergence bounds in a metric strong enough to entail guarantees about differential privacy (i.e., a multiplicative rather than additive guarantee on the probability density).

Some complications arise, since we need to repeat the privacy and optimality analysis of our mechanisms in the presence of approximation errors (such as an approximate covariance matrix and an approximate separation oracle for K). The details can be found in Section 3.7.

3.1.2 Preliminaries

Notation. We will write B_p^d to denote the unit ball of the p-norm in $\mathbb{R}^d$. When $K \subseteq \mathbb{R}^d$ is a centrally symmetric convex set, we write $\|\cdot\|_K$ for the (Minkowski) norm defined by K (i.e. $\|x\|_K = \inf\{r : x \in rK\}$). The ℓ_p-norms are denoted by $\|\cdot\|_p$, but we use $\|\cdot\|$ as a shorthand for the Euclidean norm $\|\cdot\|_2$. Given a function $F : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ and a set $K \in \mathbb{R}^{d_1}$, FK denotes the set $\{F(x) : x \in K\}$.

Differential Privacy. The definition of differential privacy is transitive in the following sense.

Fact 3.1.7. *If $M = \{\mu_x\}_{x\in\mathbb{R}^N}$ is an ε-differentially private mechanism then for measurable $S \subseteq \mathbb{R}^d$ we have $\frac{\mu_x(S)}{\mu_y(S)} \leqslant \exp(\varepsilon\|x-y\|_1)$.*

Definition 3.1.8 (Error). Let $F : \mathbb{R}^N \to \mathbb{R}^d$ and $\ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$. We denote the *$\ell_2$-error* of a mechanism M as

$$\mathrm{err}(M, F) = \sup_{x\in\mathbb{R}^N} \mathop{\mathbb{E}}_{v\sim\mu_x} \|v - Fx\|.$$

Our goal is to show trade-offs between privacy and error. We will consider linear mappings $F : \mathbb{R}^N \to \mathbb{R}^d$ which have ℓ_1-sensitivity $O(d)$, i.e., $\sup_{x\in B_1^N} \|Fx\|_1 \leqslant d$. Corollary 2.4.6 implies that for every query $F \in [-1,1]^{d\times N}$ and every $\varepsilon, \delta > 0$, there is an

— $(\varepsilon, 0)$-differentially private mechanism M with $\mathrm{err}(M, F) = O(d\sqrt{d}/\varepsilon)$.

— (ε, δ)-differentially private mechanism M with $\mathrm{err}(M, F) = O(d\sqrt{\log(1/\delta)}/\varepsilon)$.

Isotropic Position. We will use the following standard notion of *isotropic position* throughout this chapter.

Definition 3.1.9 (Isotropic Position)**.** We say a convex body $K \subseteq \mathbb{R}^d$ is in *isotropic position* with isotropic constant L_K if for every unit vector $v \in \mathbb{R}^d$,

$$\frac{1}{\mathrm{Vol}(K)} \int_K |\langle z, v \rangle|^2 dz = L_K^2 \mathrm{Vol}(K)^{2/d}. \tag{3.3}$$

Fact 3.1.10. *For every convex body $K \subseteq \mathbb{R}^d$, there is a volume-preserving linear transformation T such that TK is in isotropic position.*

For an arbitrary convex body K, its isotropic constant L_K can then be defined to be L_{TK} where T brings L to isotropic position. It is known (e.g. [MP]) that T is unique up to an orthogonal transformation and thus this is well-defined. We refer the reader to the paper of Milman and Pajor [MP], as well as the extensive survey of Giannopoulos [Gia] for a proof of this fact and other facts regarding the isotropic constant.

3.2 Lower bounds via volume estimates

In this section we show that lower bounds on the volume of the convex body $FB_1^N \subseteq \mathbb{R}^d$ give rise to lower bounds on the error that any private mechanism must have with respect to F.

Definition 3.2.1. A set of points $Y \subseteq \mathbb{R}^d$ is called a *r-packing* if $\|y - y'\|_2 \geqslant r$ for any $y, y' \in Y, y \neq y'$.

Lemma 3.2.2. *Let $K \subseteq \mathbb{R}^d$ be a measurable set. Then, K contains an $\frac{1}{4}\mathrm{vr}(K)$-packing of size* $\exp(d)$.

Proof. By the definition of $\mathrm{vr}(K)$, the set K has the same volume as a ball of radius $\mathrm{vr}(K)$. Hence, any maximal $\frac{\mathrm{vr}(K)}{4}$-packing then has the desired property. ∎

Theorem 3.2.3. *Let $\varepsilon > 0$ and suppose $F\colon \mathbb{R}^N \to \mathbb{R}^d$ is a linear map and let $K = FB_1^N$. Then, every ε-differentially private mechanism M must have*

$$\mathrm{err}(M, F) \geqslant \Omega\left(\frac{d}{\varepsilon} \cdot \mathrm{vr}(K)\right). \tag{3.4}$$

Proof. Put $\lambda = d/2\varepsilon$. By Lemma 3.2.2, λK contains an $\frac{1}{4}\lambda \mathrm{vr}(K)$-packing Y of size $\exp(d)$. Let $X \subseteq \mathbb{R}^N$ be a set of arbitrarily chosen preimages of $y \in Y$ so that $|X| = |Y|$ and $FX = Y$. By linearity, $\lambda K = F(\lambda B_1^N)$ and hence we may assume that every $x \in X$ satisfies $\|x\|_1 \leqslant \lambda$.

We will now assume that $M = \{\mu_x \colon x \in \mathbb{R}^N\}$ is an ε-differentially private mechanism with error $\frac{d}{16\varepsilon}\mathrm{vr}(K)$ and lead this to a contradiction. By the assumption on the error, Markov's inequality implies that for all $x \in X$, we have $\mu_x(B_x) \geqslant 1/2$, where B_x is a ball of radius $\frac{d}{8\varepsilon}\mathrm{vr}(K) = \frac{\lambda}{4}\mathrm{vr}(K)$ centered at Fx. Since $Y = FX$ is an $\frac{\lambda}{4}\mathrm{vr}(K)$-packing, the balls $\{B_x : x \in X\}$ are disjoint. Since $\|x\|_1 \leqslant \lambda$, it follows from ε-differential privacy with Fact 3.1.7 that

$$\mu_0(B_x) \geqslant \exp(-\varepsilon\lambda)\mu_x(B_x) \geqslant \frac{\exp(-d/2)}{2}.$$

Since the balls B_x are pairwise disjoint,

$$1 \geqslant \mu_0\left(\bigcup_{x\in X} B_x\right) = \sum_{x\in X}\mu_0(B_x) \geqslant \frac{\exp(d)\exp(-d/2)}{2} > 1 \tag{3.5}$$

for $d \geqslant 2$. We have thus obtained a contradiction. ∎

We will later need the following generalization of the previous argument which gives a lower bound in the case where K is close to a lower dimensional subspace and hence the volume inside this subspace may give a stronger lower bound.

Corollary 3.2.4. *Let $\varepsilon > 0$ and suppose $F \colon \mathbb{R}^N \to \mathbb{R}^d$ is a linear map and let $K = FB_1^N$. Furthermore, let P denote the orthogonal projection operator of a k-dimensional subspace of $\mathbb{R}^d$ for some $1 \leqslant k \leqslant d$. Then, every ε-differentially private mechanism M must have*

$$\mathrm{err}(M, F) \geqslant \Omega\left(\frac{k \cdot \mathrm{vr}_k(PK)}{\varepsilon}\right) \tag{3.6}$$

where

$$\mathrm{vr}_k \stackrel{\mathrm{def}}{=} \frac{\mathrm{Vol}_k(PK)^{1/k}}{\mathrm{Vol}(B_2^k)}.$$

Proof. Note that a differentially private answer v to Fx can be projected down to a (differentially private) answer Pv to PFx. Since P has operator norm $\|P\| \leqslant 1$ this does not increase the error, i.e.,

$$\|Pv - PFx\| \leqslant \|P(v - Fx)\| \leqslant \|P\| \cdot \|v - Fx\| \leqslant \|v - Fx\|.$$

∎

We will denote by $\mathrm{VolLB}(F, \varepsilon)$ the best lower bound obtainable in this manner, i.e.,

$$\mathrm{VolLB}(F, \varepsilon) = \sup_{k,P} \frac{k \cdot \mathrm{vr}_k(PFB_1^N)}{\varepsilon}$$

where the supremum is taken over all $k \in \{1, \ldots, d\}$ and all k-dimensional orthogonal projections P.

3.2.1 Lower bounds for small number of queries

As shown previously, the task of proving lower bounds on the error of private mechanisms reduces to analyzing the volume of FB_1^N. When $d \leqslant \log N$ this is a straightforward task.

Fact 3.2.5. *Let $d \leqslant \log N$. Then, for all matrices $F \in [-1,1]^{d \times N}$, $\mathrm{Vol}(FB_1^N)^{1/d} \leqslant O(1)$. Furthermore, there is an explicit matrix F such that FB_1^N has maximum volume.*

Proof. Clearly, FB_1^N is always contained in B_∞^d and $\mathrm{Vol}(B_\infty^d)^{1/d} = 2$. On the other hand, since $n \geqslant 2^d$, we may take F to contain all points of the hypercube $H = \{-1,1\}^d$ as its columns. In this case, $FB_1^N \supseteq B_\infty^d$. ∎

This lower bound shows that the Laplacian mechanism (Corollary 2.4.6) is, in fact, optimal when $d \leqslant \log N$.

3.3 The K-norm mechanism

In this section we describe a new differentially private mechanism, which we call the K-norm mechanism.

Definition 3.3.1 (K-norm mechanism)**.** Given a linear map $F\colon \mathbb{R}^N \to \mathbb{R}^d$ and $\varepsilon > 0$, we let $K = FB_1^N$ and define the mechanism $\mathrm{KM}(F, d, \varepsilon) = \{\mu_x \colon x \in \mathbb{R}^N\}$ so that each measure μ_x is given by the probability density function

$$f(v) = Z^{-1} \exp(-\varepsilon \|Fx - v\|_K) \tag{3.7}$$

defined over $\mathbb{R}^d$. Here Z denotes the normalization constant

$$Z = \int_{\mathbb{R}^d} \exp(-\varepsilon \|Fx - v\|_K)\, \mathrm{d}v = \Gamma(d+1)\mathrm{Vol}(\varepsilon^{-1}K).$$

A more concrete view of the mechanism is provided by Figure 3.2 and justified in the next remark.

Remark 3.3.2. We can sample from the distribution μ_x as follows:

Input: Query $F \in \mathbb{R}^{d\times N}$, histogram $x \in \mathbb{R}^N$, privacy parameter $\varepsilon > 0$

1. Sample $z \in \mathbb{R}^d$ uniformly at random from $K = FB_1^N$
2. Sample $r \in \mathbb{R}$ from $\mathrm{Gamma}(d+1, \varepsilon^{-1})$

Output: $Fx + rz$

Figure 3.2: Description of the d-dimensional K-norm mechanism.

1. Sample r from the Gamma distribution with parameter $d+1$ and scale ε^{-1}, denoted $\mathrm{Gamma}(d+1, \varepsilon^{-1})$. That is, r is distributed as

$$\mathbb{P}\{r > R\} = \frac{1}{\varepsilon^{-(d+1)}\Gamma(d+1)} \int_R^\infty e^{-\varepsilon t} t^d \, \mathrm{d}t.$$

2. Sample v uniformly from $Fx + rK$.

Indeed, if $\|v - Fx\|_K = R$, then the distribution of v as above follows the probability density function

$$g(v) = \frac{1}{\varepsilon^{-(d+1)}\Gamma(d+1)} \int_R^\infty \frac{e^{-\varepsilon t} t^d}{\mathrm{Vol}(tK)} \, \mathrm{d}t = \frac{\int_R^\infty e^{-\varepsilon t} \, \mathrm{d}t}{\varepsilon^{-1}\Gamma(d+1)\mathrm{Vol}(\varepsilon^{-1}K)} = \frac{e^{-\varepsilon R}}{\Gamma(d+1)\mathrm{Vol}(\varepsilon^{-1}K)}, \tag{3.8}$$

where we used the fact that $\int_0^\infty e^{-\varepsilon t} \, \mathrm{d}t = \varepsilon^{-1}$. We thus see that this calculation is in agreement with (3.7). That is, $g(v) = f(v)$.

The next theorem shows that the K-norm mechanism is indeed differentially private. Moreover, we can express its error in terms of the *expected distance from the origin* of a random point in K.

Theorem 3.3.3. *Let $\varepsilon > 0$. Suppose $F\colon \mathbb{R}^N \to \mathbb{R}^d$ is a linear map and put $K = FB_1^N$. Then, the mechanism $KM(F, d, \varepsilon)$ is ε-differentially private, and for every $p > 0$ achieves the error bound*

$$\mathop{\mathbb{E}}_{v\sim\mu_x} \|Fx - v\|_2^p \leqslant \frac{\Gamma(d+1+p)}{\varepsilon^p \Gamma(d)} \mathop{\mathbb{E}}_{z\in K} \|z\|_2^p. \tag{3.9}$$

In particular, the ℓ_2-error is at most

$$\frac{d+1}{\varepsilon} \mathop{\mathbb{E}}_{z\in K} \|z\|_2 = \frac{d+1}{\varepsilon} \cdot \mathrm{mr}(K).$$

Proof. To argue the error bound, we will follow Remark 3.3.2. Let $D = \mathrm{Gamma}(d+1, 1/\varepsilon)$. For all $x \in \mathbb{R}^N$,

$$\mathop{\mathbb{E}}_{v\sim\mu_x} \|Fx - v\|^p = \mathop{\mathbb{E}}_{v\sim\mu_0} \|v\|^p = \mathop{\mathbb{E}}_{r\sim D} \mathop{\mathbb{E}}_{v\in rK} \|v\|^p = \left[\mathop{\mathbb{E}}_{r\sim D} r^p\right] \mathop{\mathbb{E}}_{z\in K} \|z\|^p$$
$$= \frac{\Gamma(d+1+p)}{\varepsilon^p \Gamma(d+1)} \mathop{\mathbb{E}}_{z\in K} \|z\|^p. \qquad \text{(by Fact 2.10.5)}$$

When $p = 1$, $\frac{\Gamma(d+1+p)}{\Gamma(d+1)} = d+1$.

Privacy follows from the fact that the mechanism is a special case of the exponential mechanism [MT]. For completeness, we repeat the argument. Indeed, suppose that $\|x\|_1 \leqslant 1$. It suffices to show that for all $v \in \mathbb{R}^d$, the densities of μ_0 and μ_x are within multiplicative $\exp(\varepsilon)$, i.e.,

$$\frac{Z^{-1} e^{-\varepsilon\|v\|_K}}{Z^{-1} e^{-\varepsilon\|Fx-v\|_K}} = e^{\varepsilon(\|Fx-v\|_K - \|v\|_K)} \leqslant e^{\varepsilon\|Fx\|_K} \leqslant e^{\varepsilon}.$$

where in the first inequality we used the triangle inequality for $\|\cdot\|_K$. In the second step we used that $x \in B_1^N$ and hence $Fx \in FB_1^N = K$ which means $\|Fx\|_K \leqslant 1$. Hence, the mechanism satisfies ε-differential privacy. ∎

3.4 Optimality for random queries and isotropic bodies

In this section, we will show that our upper bound matches our lower bound when F is a random query. A key observation is that FB_1^N is the *symmetric* convex hull of N (random) points $\{v_1, \ldots, v_n\} \subseteq \mathbb{R}^d$, i.e., the convex hull of $\{\pm v_1, \ldots, \pm v_n\}$, where $v_i \in \mathbb{R}^d$ is the ith column of F. The symmetric convex hull of random points has been studied extensively in the theory of random polytopes. A recent result of Litvak, Pajor, Rudelson and Tomczak-Jaegermann [LPRTJ] gives the following lower bound on the volume of the convex hull. For convenience, we state their result in terms of volume radius.

Theorem 3.4.1 ([LPRTJ]). *Let $2d \leqslant n \leqslant 2^d$ and let F denote a random $d \times N$ Bernoulli matrix. Then,*

$$\mathrm{vr}\left(FB_1^N\right) \geqslant \Omega(1)\sqrt{\log\left(\frac{N}{d}\right)}, \tag{3.10}$$

with probability $1 - \exp(-\Omega(d^\beta n^{1-\beta}))$ for any $\beta \in (0, \frac{1}{2})$. Furthermore, there is an explicit construction of n points in $\{-1, 1\}^d$ whose convex hull achieves the same volume.

We are mostly interested in the range where $N \gg d \log d$ in which case the theorem was already proved by Giannopoulos and Hartzoulaki [GH] (up to a weaker bound in the probability and without the explicit construction).

The bound in (3.10) is tight up to constant factors. A well known result [BF] shows that if K is the convex hull of any N points on the sphere in $\mathbb{R}^d$ of radius $\sqrt{d}$, then

$$\mathrm{vr}(K) \leqslant O(1)\sqrt{\log\left(\frac{N}{d}\right)}. \tag{3.11}$$

Notice, that in our case $K = FB_1^N \subseteq B_\infty^d \subseteq \sqrt{d}B_2^d$ and in fact the vertices of K are points on the $(d-1)$-dimensional sphere of radius $\sqrt{d}$. However, equation (3.10) states that the normalized volume of the random polytope K will be proportional to the volume of the Euclidean ball of radius $\sqrt{\log(N/d)}$ rather than $\sqrt{d}$. When $d \gg \log n$, this means that the volume of K will be tiny compared to the volume of the infinity ball B_∞^d. By combining the volume lower bound with Theorem 3.2.3, we get the following lower bound on the error of private mechanisms.

Theorem 3.4.2. *Let $\varepsilon > 0$ and $0 < d \leqslant N/2$. Then, for almost all matrices $F \in \{-1,1\}^{d\times N}$, every ε-differentially private mechanism M must have*

$$\mathrm{err}(M,F) \geqslant \Omega(d/\varepsilon) \cdot \min\left\{\sqrt{d}, \sqrt{\log\left(\frac{N}{d}\right)}\right\}. \tag{3.12}$$

3.4.1 A separation result

We use this paragraph to point out that our lower bound immediately implies a separation between (ε,δ)-differential privacy and $(\varepsilon,0)$-differential privacy. On the other hand, Corollary 2.4.6 gives (ε,δ)-differential privacy with error $o\left(\varepsilon^{-1}\sqrt{\log(N/d)}\right)$ as long as $\delta \geqslant 1/n^{o(1)}$. Our lower bound in Theorem 3.4.2 on the other hand states that the error of any ε-differentially private mechanism must be $\Omega\left(\varepsilon^{-1}\sqrt{\log(N/d)}\right)$ (assuming $d \gg \log(n)$). We get the strongest separation when $d \leqslant \log(n)$ and δ is constant. In this case, our lower bound is a factor $\sqrt{d}$ larger than the upper bound for approximate differential privacy.

3.4.2 Upper bound on average Euclidean norm

Klartag and Kozma [KK] recently gave a bound on the quantity $\mathbb{E}_{z\in K}\|z\|$ when $K = FB_1^N$ for random F.

Theorem 3.4.3 ([KK])**.** *Let F be a random $d \times N$ Bernoulli matrix and put $K = FB_1^N$. Then, there is a constant $C > 0$ so that with probability greater than $1 -$*

$Ce^{-O(n)}$,

$$\frac{1}{\mathrm{Vol}(K)} \int_{z \in K} \|z\|^2 \, \mathrm{d}z \leqslant C \log\left(\frac{N}{d}\right). \tag{3.13}$$

An application of Jensen's inequality thus gives us the following corollary.

Corollary 3.4.4. *Let $\varepsilon > 0$ and $0 < d \leqslant N/2$. Then, for almost all matrices $F \in \{-1,1\}^{d \times N}$, the mechanism $KM(F, d, \varepsilon)$ is ε-differentially private with error at most*

$$O\left(\frac{d}{\varepsilon}\right) \cdot \min\left\{\sqrt{d}, \sqrt{\log\left(\frac{N}{d}\right)}\right\}. \tag{3.14}$$

3.5 Approximately isotropic bodies

The following definition is a relaxation of isotropic position used in literature (e.g., [KLS])

Definition 3.5.1 (Approximately Isotropic Position)**.** We say a convex body $K \subseteq \mathbb{R}^d$ is in *c-approximately isotropic position* if for every unit vector $v \in \mathbb{R}^d$,

$$\frac{1}{\mathrm{Vol}(K)} \int_K |\langle z, v \rangle|^2 \, \mathrm{d}z \leqslant c^2 L_K^2 \mathrm{Vol}(K)^{\frac{2}{d}}. \tag{3.15}$$

The results of Klartag and Kozma [KK] referred to in the previous section show that the symmetric convex hull of n random points from the d-dimensional hypercube are in $O(1)$-approximately isotropic position and have $L_K = O(1)$. More generally, the K-norm mechanism can be shown to be approximately optimal whenever K is nearly isotropic.

Theorem 3.5.2 (Theorem 3.1.2 restated)**.** *Let $\varepsilon > 0$. Suppose $F \colon \mathbb{R}^N \to \mathbb{R}^d$ is a linear map such that $K = FB_1^N$ is in c-approximately isotropic position. Then, the K-norm mechanism is ε-differentially private and has error at most $O(cL_K) \cdot \frac{d\mathrm{vr}(K)}{\varepsilon}$*

Proof. By Theorem 3.3.3, the K-norm mechanism is ε-differentially private and has error $\frac{d+1}{\varepsilon} \mathbb{E}_{z \in K} \|z\|$. By the definition of the approximately isotropic position, we have: $\mathbb{E}_{z \in K} \|z\|^2 \leqslant d \cdot c^2 L_K^2 \mathrm{Vol}(K)^{2/d}$. By Jensen's inequality,

$$\frac{d+1}{\varepsilon} \mathop{\mathbb{E}}_{z \in K} \|z\| \leqslant \frac{d+1}{\varepsilon} \sqrt{\mathop{\mathbb{E}}_{z \in K} \|z\|^2} \leqslant O\left(\frac{cL_K d\sqrt{d}\mathrm{Vol}(K)^{1/d}}{\varepsilon}\right) = O\left(\frac{cL_K d\mathrm{vr}(K)}{\varepsilon}\right).$$

■

We can see that the previous upper bound is tight up to a factor of cL_K. Estimating L_K for general convex bodies is a well-known open problem in convex geometry. The best known upper bound for a general convex body $K \subseteq \mathbb{R}^d$ is $L_K \leqslant O(d^{1/4})$ due to Klartag [Kla], improving over the estimate $L_K \leqslant O(d^{1/4} \log d)$ of Bourgain from '91. The conjecture is that $L_K = O(1)$.

Conjecture 3.5.1 (Hyperplane Conjecture). *There exists $C > 0$ such that for every d and every convex set $K \subseteq \mathbb{R}^d$, $L_K < C$.*

Assuming this conjecture we get matching bounds for approximately isotropic convex bodies.

Theorem 3.5.3. *Let $\varepsilon > 0$. Assuming the hyperplane conjecture, for every $F \in [-1,1]^{d \times N}$ such that $K = FB_1^N$ is c-approximately isotropic, the K-norm mechanism $KM(F, d, \varepsilon)$ is ε-differentially private with error at most*

$$O\left(\frac{cd}{\varepsilon}\right) \cdot \min\left\{\sqrt{d}, \sqrt{\log\left(\frac{N}{d}\right)}\right\}. \tag{3.16}$$

3.6 Non-isotropic bodies

While the mechanism of the previous sections is near-optimal for near-isotropic queries, it can be far from optimal if K is far from isotropic. For example, suppose the matrix F has random entries from $\{+1, -1\}$ in the first row, and (say) from $\{\frac{1}{d^2}, -\frac{1}{d^2}\}$ in the remaining rows. While the Laplacian mechanism will add $O(\frac{1}{\varepsilon})$ noise to the first co-ordinate of Fx, the K-norm mechanism will add noise $O(d/\varepsilon)$ to the first co-ordinate. Moreover, the volume lower bound VolLB is at most $O(\varepsilon^{-1}\sqrt{d})$. Rotating F by a random rotation gives, w.h.p., a query for which the Laplacian mechanism adds ℓ_2 error $O(d/\varepsilon)$. For such a body, the Laplacian and the K-norm mechanisms, as well as the VolLB are far from optimal.

In this section, we will design a recursive mechanism that can handle such non-isotropic convex bodies. To this end, we will need to introduce a few more notions from convex geometry.

Suppose $K \subseteq \mathbb{R}^d$ is a centered convex body, i.e. $\int_K x \, dx = 0$. The *covariance matrix of K*, denoted M_K is the $d \times d$ matrix with entry ij equal to $M_{ij} = \frac{1}{\mathrm{Vol}(K)} \int_K x_i x_j \, dx$. That is, M_K is the covariance matrix of the uniform distribution over K.

3.6.1 A recursive mechanism

Having defined the covariance matrix, we can describe a recursive mechanism for the case when K is not in isotropic position. The idea of the mechanism is

NiKM(F, d, ε):

1. Let $K = FB_1^N$. Let $\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_d$ denote the eigenvalues of the covariance matrix M_K. Pick a corresponding orthonormal eigenbasis $u_1, \ldots, u_d$.
2. Let $d' = \lfloor d/2 \rfloor$ and let $U = \mathrm{span}\{u_1, \ldots, u_{d'}\}$ and $V = \mathrm{span}\{u_{d'+1}, \ldots, v_d\}$.
3. Sample $v \sim \mathrm{KM}(F, d, \varepsilon)$.
4. If $d = 1$, output $P_V v$. Otherwise, output $\mathrm{NiKM}(P_U F, d', \varepsilon) + P_V v$.

Figure 3.3: Mechanism for non-isotropic bodies

to act differently on different eigenspaces of the covariance matrix. Specifically, the mechanism will use a lower-dimensional version of $\mathrm{KM}(F, d', \varepsilon)$ on subspaces corresponding to few large eigenvalues.

Our mechanism, called $\mathrm{NiKM}(F, d, \varepsilon)$, is given a linear mapping $F\colon \mathbb{R}^N \to \mathbb{R}^d$, and parameters $d \in \mathbb{N}, \varepsilon > 0$. The mechanism proceeds recursively by partitioning the convex body K into two parts defined by the middle eigenvalue of M_K. On one part it will act according to the K-norm mechanism. On the other part, it will descend recursively. The mechanism is described in Figure 3.3

Remark 3.6.1. The image of $P_U F$ above is a d'-dimensional subspace of $\mathbb{R}^d$. We assume that in the recursive call $\mathrm{NiKM}(P_U F, d', \varepsilon)$, the K-norm mechanism is applied to a basis of this subspace. However, formally the output is a d-dimensional vector.

To analyze our mechanism, first observe that the recursive calls terminate after at most $\log d$ steps. For each recursive step $m \in \{0, \ldots, \log d\}$, let v_m denote the distribution over the output of the K_m-norm mechanism in step 3. Here, K_m denotes the d_m-dimensional body given in step m.

Lemma 3.6.2. *The mechanism NiKM(F, d, ε) satisfies $(\varepsilon \log d)$-differential privacy.*

Proof. We claim that for every step $m \in \{0, \ldots, \log d\}$, the distribution over v_m is ε-differentially private. Notice that this claim implies the lemma, since the joint distribution of $v_0, v_1, \ldots, v_m$ is $\varepsilon \log(d)$-differentially private. In particular, this is true for the final output of the mechanism as it is a function of $v_0, \ldots, v_m$.

To see why the claim is true, observe that each K_m is the d_m-dimensional image of the ℓ_1-ball under a linear mapping. Hence, the K_m-norm mechanism guarantees ε-differential privacy by Theorem 3.3.3. ∎

The error analysis of our mechanism requires more work. In particular, we need to understand how the volume of $P_U K$ compares to the norm of $P_V v$. As a first step we will analyze the volume of $P_U K$.

3.6.2 Volume in eigenspaces of the covariance matrix

Our goal in this section is to express the volume of K in eigenspaces of the covariance matrix in terms of the eigenvalues of the covariance matrix. This will be needed in the analysis of our mechanism for non-isotropic bodies.

We start with a formula for the volume of central sections of isotropic bodies. This result can be found in [MP].

Proposition 3.6.3. *Let $K \subseteq \mathbb{R}^d$ be an isotropic body of unit volume. Let E denote a k-dimensional subspace for $1 \leqslant k \leqslant d$. Then,*

$$\mathrm{Vol}_k(E \cap K)^{1/(d-k)} = \Theta\left(\frac{L_{B_K}}{L_K}\right).$$

Here, B_K is an explicitly defined isotropic convex body.

From here on, for an isotropic body K, let $\alpha_K = \Omega(L_{B_K}/L_K)$ be a lower bound on $\mathrm{Vol}_k(E \cap K)^{1/(d-k)}$ implied by the above proposition. For a non-isotropic K, let α_K be α_{TK} when T is the map that brings K into isotropic position. Notice that if the Hyperplane Conjecture is true, then $\alpha_K = \Omega(1)$. Moreover, α_K is $\Omega(d^{\frac{1}{4}})$ due to the results of [Kla].

Corollary 3.6.4. *Let $K \subseteq \mathbb{R}^d$ be an isotropic body with $\mathrm{Vol}(K) = 1$. Let E denote a k-dimensional subspace for $1 \leqslant k \leqslant d$ and let P denote an orthogonal projection operator onto the subspace E. Then,*

$$\mathrm{Vol}_k(PK)^{1/(d-k)} \geqslant \alpha_K.$$

Proof. Observe that PK contains $E \cap K$ since P is the identity on E. ∎

We cannot immediately use these results since they only apply to isotropic bodies and we are specifically dealing with non-isotropic bodies. The trick is to apply the previous results after transforming K into an isotropic body while keeping track how much this transformation changed the volume.

As a first step, the following lemma relates the volume of projections of an arbitrary convex body K to the volume of projections of TK for some linear mapping T.

Lemma 3.6.5. *Let $K \subseteq \mathbb{R}^d$ be a symmetric convex body. Let T be a linear map which has eigenvectors $u_1, \ldots, u_d$ with eigenvalues $\lambda_1, \ldots, \lambda_d$. Let $1 \leqslant k \leqslant d$ and suppose $E = \mathrm{span}\{u_1, u_2, \ldots, u_k\}$, Denote by P be the projection operator onto the subspace E. Then,*

$$\mathrm{Vol}_k(PK) \geqslant \mathrm{Vol}_k(PTK) \prod_{i=1}^{k} \lambda_i^{-1}.$$

Proof. For simplicity, we assume that the eigenvectors of T are the standard basis vectors $e_1, \ldots, e_d$; this is easily achieved by applying a rotation to K. Now, it is easy to verify that $P = PT^{-1}T = SPT$ where $S = \mathrm{diag}(\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_k^{-1}, 0, \ldots, 0)$. Thus we can write

$$\mathrm{Vol}_k(PK) = \det(S_{|E})\mathrm{Vol}_k(PTK) = \frac{1}{\prod_{i=1}^{k} \lambda_i} \mathrm{Vol}_k(PTK). \qquad \blacksquare$$

Before we can finish our discussion, we will need the fact that the isotropic constant of K can be expressed in terms of the determinant of M_K.

Fact 3.6.6 ([Gia, MP])**.** *Let $K \subseteq \mathbb{R}^d$ be a convex body of unit volume. Then,*

$$L_K^2 \mathrm{Vol}(K)^{\frac{2}{d}} = \det(M_K)^{1/d}. \tag{3.17}$$

Moreover, K is in isotropic position if and only if $M_K = L_K^2 \mathrm{Vol}(K)^{2/d} I$.

We conclude with the following Proposition 3.6.7.

Proposition 3.6.7. *Let $K \subseteq \mathbb{R}^d$ be a symmetric convex body. Let M_K have eigenvectors $u_1, \ldots, u_d$ with eigenvalues $\sigma_1, \ldots, \sigma_d$. Let $1 \leqslant k \leqslant \lceil \frac{d}{2} \rceil$ with and suppose $E = \mathrm{span}\{u_1, u_2, \ldots, u_k\}$. Denote by P be the projection operator onto the subspace E. Then,*

$$\mathrm{Vol}_k(PK)^{1/(d-k)} \geqslant \Omega(1) \cdot \alpha_K \left(\prod_{i=1}^{k} \sigma_i^{1/2} \right)^{1/(d-k)}, \tag{3.18}$$

where α_K is $\Omega(1/d^{\frac{1}{4}})$. Moreover, assuming the Hyperplane conjecture, $\alpha_K \geqslant \Omega(1)$.

Proof. Consider the linear mapping $T = M_K^{-1/2}$. this is well defined since M_K is a positive symmetric matrix. It is easy to see that after applying T, we have $M_{TK} = I$. Hence, by Fact 3.6.6, TK is in isotropic position and has volume $\mathrm{Vol}(TK)^{1/d} = 1/L_{TK} = 1/L_K$, since $\det(M_{TK}) = 1$. Scaling TK by $\lambda = L_K^{1/d}$ hence results in $\mathrm{Vol}(\lambda TK) = 1$. Noting that λT has eigenvalues $\lambda\sigma_1^{-\frac{1}{2}}, \lambda\sigma_2^{-\frac{1}{2}}, \ldots, \lambda\sigma_d^{-\frac{1}{2}}$, we can apply Lemma 3.6.5 and get

$$\mathrm{Vol}_k(PK) \geqslant \mathrm{Vol}_k(P\lambda TK) \prod_{i=1}^{k} \frac{\sqrt{\sigma_i}}{\lambda}$$

Since λTK is in isotropic position and has unit volume, Corollary 3.6.4 implies that

$$\mathrm{Vol}_k(P_{\lambda TK})^{1/(d-k)} \geqslant \alpha_K . \tag{3.19}$$

Thus the required inequality holds with an additional $\lambda^{-\frac{k}{d-k}}$ term. By assumption on k, $\frac{k}{d-k}$ is at most 2. Moreover, $\lambda = L_K^{1/d} \leqslant d^{1/d} \leqslant 2$, so that this additional term is a constant. As discussed above, α_K is $\Omega(d^{-\frac{1}{4}})$ by [Kla], and $\Omega(1)$ assuming the Hyperplane Conjecture 3.5.1. Hence the claim. ∎

3.6.3 Arguing near optimality of our mechanism

Our next lemma shows that the expected squared Euclidean error added by our algorithm in each step is bounded by the square of the optimum. We will first need the following fact.

Fact 3.6.8. *Let $K \subseteq \mathbb{R}^d$ be a centered convex body. Let $\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_d$ denote the eigenvalues of M_K with a corresponding orthonormal eigenbasis $u_1, \ldots, u_d$. Then, for all $1 \leqslant i \leqslant d$,*

$$\sigma_i = \max_{\theta} \mathop{\mathbb{E}}_{x \in K} \langle \theta, x \rangle^2 \tag{3.20}$$

where the maximum runs over all $\theta \in \mathbb{S}^{d-1}$ such that θ is orthogonal to $u_1, u_2, \ldots, u_{i-1}$.

Lemma 3.6.9. *Let v denote the random variable returned by the K-norm mechanism in step (3) in the above description of NiKM(F, d, ε). Then,*

$$\mathrm{VolLB}(F, \varepsilon)^2 \geqslant \Omega(\alpha_K^2)\, \mathbb{E}\|P_V v\|_2^2 .$$

Proof. For simplicity, we will assume that d is even and hence $d - d' = d'$. The analysis of the K-norm mechanism (Theorem 3.3.3 with $p = 2$) shows that the random variable v returned by the K-norm mechanism in step (3) satisfies

$$\begin{aligned}
\mathbb{E}\|P_V v\|_2^2 = \frac{\Gamma(d+3)}{\varepsilon^2 \Gamma(d+1)} &= \frac{(d+2)(d+1)}{\varepsilon^2} \mathop{\mathbb{E}}_{z \in K} \|P_V z\|_2^2 \\
&= O\left(\frac{d^2}{\varepsilon^2}\right) \sum_{i=d'+1}^{d} \mathop{\mathbb{E}}_{z \in K} \langle z, u_i \rangle^2 \\
&= O\left(\frac{d^2}{\varepsilon^2}\right) \sum_{i=d'+1}^{d} \sigma_i \qquad \text{(by Fact 3.6.8)} \\
&\leqslant O\left(\frac{d^3}{\varepsilon^2}\right) \cdot \sigma_{d'+1} . \qquad (3.21)
\end{aligned}$$

On the other hand, by the definition of VolLB,

$$\begin{aligned}\mathrm{VolLB}(F,\varepsilon)^2 &\geqslant \Omega\left(\frac{d^3}{\varepsilon^2}\right)\cdot \mathrm{Vol}_{d'}(P_U K)^{2/d'}\\ &\geqslant \Omega\left(\frac{d^3}{\varepsilon^2}\right)\alpha_K^2\left(\prod_{i=1}^{d'}\sigma_i\right)^{1/d'} && \text{(by Proposition 3.6.7)}\\ &\geqslant \Omega\left(\frac{d^3}{\varepsilon^2}\right)\alpha_K^2\sigma_{d'}.\end{aligned}$$

Since $\sigma_{d'} \geqslant \sigma_{d'+1}$, it follows that

$$\mathrm{VolLB}(F,\varepsilon)^2 \geqslant \Omega(\alpha_K^2)\,\mathbb{E}\|P_V a\|^2. \tag{3.22}$$

The case of odd d is similar except that we define K' to be the projection onto the first $d'+1$ eigenvectors. ∎

Lemma 3.6.10. *Assume the Hyperplane Conjecture. Then, the ℓ_2-error of the mechanism NiKM(F,d,ε) satisfies*

$$\mathrm{err}(\mathit{NiKM},F) \leqslant O\Big(\sqrt{\log(d)}\cdot \mathrm{VolLB}(F,\varepsilon)\Big).$$

Proof. We have to sum up the error over all recursive calls of the mechanism. To this end, let $P_{V_m}v_m$ denote the output of the K-norm mechanism v_m in step m projected to the corresponding subspace V_m. Also, let $v \in \mathbb{R}^d$ denote the final output of our mechanism. We then have,

$$\begin{aligned}\mathbb{E}\|v\|_2 &\leqslant \sqrt{\mathbb{E}\|v\|_2^2} && \text{(Jensen's inequality)}\\ &= \sqrt{\sum_{m=1}^{\log d}\mathbb{E}\|P_{V_m}v_m\|_2^2}\\ &\leqslant \sqrt{\sum_{m=1}^{\log d} O(\alpha_{K_m}^{-2})\cdot \mathrm{VolLB}(F,\varepsilon)^2} && \text{(by Lemma 3.6.9)}\\ &\leqslant O(\sqrt{\log d})\Big(\max_m \alpha_{K_m}^{-1}\Big)\mathrm{VolLB}(F,\varepsilon).\end{aligned}$$

Here we have used the fact that $\mathrm{VolLB}(F,\varepsilon) \geqslant \mathrm{VolLB}(P_U F,\varepsilon)$. Finally, the hyperplane conjecture implies $\max_m \alpha_{K_m}^{-1} = O(1)$. ∎

Corollary 3.6.11. *Let $\varepsilon > 0$. Suppose $F\colon \mathbb{R}^N \to \mathbb{R}^d$ is a linear map. Further, assume the hyperplane conjecture. Then, there is an ε-differentially private mechanism M with error*

$$\mathrm{err}(M,F) \leqslant O(\log(d)^{3/2}\cdot \mathrm{VolLB}(F,\varepsilon)).$$

Proof. The mechanism NiKM$(F, d, \varepsilon/\log(d))$ satisfies ε-differential privacy, by Lemma 3.6.2. The error is at most $\log(d)\sqrt{\log d} \cdot \mathrm{VolLB}(F, \varepsilon)$ as a direct consequence of Lemma 3.6.10. ∎

Thus our lower bound VolLB and the mechanism NiKM are both within $O(\log^{3/2} d)$ of the optimum.

3.7 More efficient implementation using geometric random walks

We will first describe how to implement our basic mechanism KM(F, d, ε). As we saw, this mechanism is optimal when FB_1^N is in roughly isotropic position. In Section 3.7.1, we extend our discussion to NiKM(F, d, ε) thus getting an efficient nearly optimal mechanism even when FB_1^N is not in isotropic position.

Recall that we first sample $r \sim \mathrm{Gamma}(d+1, \varepsilon^{-1})$ and then sample a point v uniformly at random from rK. The first step poses no difficulty. Indeed, when $U_1, \ldots, U_d$ are independently distributed uniformly over the interval $(0, 1]$, then a standard fact tells us that

$$\frac{1}{\varepsilon} \sum_{i=1}^{d+1} -\ln(U_i) \sim \mathrm{Gamma}(d+1, \varepsilon^{-1}).$$

Sampling uniformly from K on the other hand may be hard. However, there are ways of sampling nearly uniform points from K using various types of rapidly mixing random walks. In this section, we will use the *Grid Walk* for simplicity even though there are more efficient walks that will work for us. We refer the reader to the survey of Vempala [Vem] or the original paper of Dyer, Frieze and Kannan [DFK] for a description of the Grid walk and background information. Informally, the Grid walk samples nearly uniformly from a grid inside K, i.e., $\mathcal{L} \cap K$ where we take $\mathcal{L} = \frac{1}{d^2}\mathbb{Z}^d$. The Grid Walk poses two requirements on K:

1. Membership in K can be decided efficiently.
2. K is bounded, in the sense that $B_2^d \subseteq K \subseteq dB_2^d$.

Both conditions are naturally satisfied in our case where $K = FB_1^N$ for some $F \in [-1, 1]^{d \times N}$. Indeed, $K \subseteq B_\infty^d \subseteq \sqrt{d} B_2^d$ and we assume throughout this section that $B_2^d \subseteq K$. This is without loss of generality, since we may replace K by $K' = K + B_2^d$. This will only increase the noise level by 1 in Euclidean distance. Notice that K' is convex.

The exact notion of membership oracle that we need is given in the next definition.

Definition 3.7.1. A *β-weak* separation oracle for K is a blackbox that says 'YES' when given $u \in \mathbb{R}^d$ with $(u + \beta B_2^d) \subseteq K$ and outputs 'NO' when $u \notin K + \beta B_2^d$.

In order to implement a weak membership oracle for K, we need to be able to decide for a given $v \in \mathbb{R}^d$, whether there exists an $x \in B_1^N$ such that $Fx = v$. These constraints can be encoded using a linear program. In the case of K' this can be done using standard convex programming techniques [GLS].

Lemma 3.7.2. *Let $\beta > 0$. We can implement a β-weak separation oracle for K and also $K + B_2^d$ in time polynomial in $N, d, 1/\beta$.*

The mixing time of the Grid walk is usually quantified in terms of the total variation (or L_1) distance between the random walk and its stationary distribution. The stationary distribution of the grid Walk is the uniform distribution over $\mathcal{L} \cap K$. Standard arguments show that an L_1-bound gives us (ε, δ)-differential privacy where δ can be made exponentially small in polynomial time. In order to get $(\varepsilon, 0)$-differential privacy we instead need a multiplicative guarantee on the density of the random walk at each point in K.

It is not difficult to show that the Grid Walk actually satisfies mixing bounds in a pointwise multiplicative sense. We also need to take care of the fact that the stationary distribution is a priori not uniform over K.

Theorem 3.7.3. *There is a mechanism M' with expected runtime polynomial in N, d and ε^{-1} such that*

1. *M' is ε-differentially private,*
2. $\mathrm{err}(M', F) = O(\mathrm{err}(KM(F, d, \varepsilon), F))$.

To prove the theorem we need the next lemma that essentially directly follows from [DFK].

Lemma 3.7.4. *There is a randomized algorithm* $\mathrm{Sample}(K, \beta)$ *running in time* $\mathrm{poly}(N, d, \beta^{-1})$ *whose output distribution is pointwise within a* $(1 \pm \beta)$ *factor from the uniform distribution over a body $\widehat{K}$ such that $K \subseteq \widehat{K} \subseteq (1 + \beta)K$.*

Proof sketch. In order to implement $\mathrm{Sample}(K, \beta)$ we consider the t-step grid walk over K using a $\beta/2$-weak separation oracle and a fine enough grid $\mathcal{L} = \frac{\beta}{2}\mathbb{Z}^d \cap dB_2^d$ where $\beta = \mathrm{poly}(1/d)$. By Lemma 3.7.2, we can implement the separation oracle in time $\mathrm{poly}(N, d, \beta^{-1})$.

It is known [DFK] that the t-step Grid Walk for $t = \mathrm{poly}(d, 1/\beta, \log(1/\Delta))$ gets within statistical distance at most Δ of the uniform distribution over $\mathcal{L} \cap K'$ where K' is a body satisfying $(1-\beta/2)K \subseteq K' \subseteq (1+\beta/2)K$,

Setting Δ to be much smaller than the number of atoms in the Grid Walk, i.e., $\Delta = \mathrm{poly}(\varepsilon\beta/|\mathcal{L}|)$ we end up with a distribution that is point-wise within at $(1 \pm \beta)$-factor of the uniform distribution over $\mathcal{L} \cap K'$. Note that $\log(1/\Delta) = \mathrm{poly}(d, 1/\varepsilon, 1/\beta)$.

Let Z be a sample from the grid walk described above, and let $\widehat{Z}$ be a random point from an ℓ_∞-ball of radius $\beta/4$ centered at Z. Then $\widehat{Z}$ is a nearly uniform sample from a body $\widehat{K}$ which has the property that $(1-\beta)K \subseteq \widehat{K} \subseteq (1+\beta)K$. ∎

Proof of Theorem 3.7.3. Our algorithm first samples $r \sim \mathrm{Gamma}(d+1, 10\varepsilon^{-1})$, and then outputs $Fx + rz$ where z is the output of $\mathrm{Sample}(K, \beta)$ for $\beta = \min\{\varepsilon/10d, 1/10r\}$. Let g denote the resulting density function of our mechanism. Let $v \in \mathbb{R}^d$. We will compare $g(v)$ to the density $f(v)$ of $\mathrm{KM}(F, d, \varepsilon/10)$.

We can repeat the calculation for the density at a point v in equation (3.8). Indeed for a point v with $\|v - Fx\|_K = R$, the density at v conditioned on a sample r from the Gamma distribution, is $(1 \pm \beta)/\mathrm{Vol}(r\widehat{K})$ whenever $v \in r\widehat{K}$, and zero otherwise. By our choice of β, $\mathrm{Vol}(\widehat{K}) = (1 \pm \varepsilon/5)\mathrm{Vol}(K)$. Moreover, since $K \subseteq \widehat{K} \subseteq (1+\beta)K$ we have $v \in r\widehat{K}$ for $r \geqslant R$. Thus the density at v is

$$g(v) \geqslant \frac{1-\varepsilon/5}{\varepsilon^{-d}\Gamma(d+1)} \int_R^\infty \frac{e^{-\varepsilon t} t^d}{\mathrm{Vol}(tK)}\,\mathrm{d}t \geqslant \frac{\exp(-\varepsilon/2)e^{-\varepsilon R}}{\Gamma(d+1)\mathrm{Vol}(\varepsilon^{-1}K)} = \exp(-\varepsilon/2) f(v).$$

Similarly, using the fact that $v \notin r\widehat{K}$ for $r < R/(1+\beta)$. we have

$$g(v) \leqslant \frac{\exp(\varepsilon/5)}{\varepsilon^{-d}\Gamma(d+1)} \int_{R/(1+\beta)}^\infty \frac{e^{-\varepsilon t} t^d}{\mathrm{Vol}(tK)}\,\mathrm{d}t \leqslant \frac{\exp(\varepsilon/5)e^{-\varepsilon R/(1+\beta)}}{\Gamma(d+1)\mathrm{Vol}(\varepsilon^{-1}K)}$$

Since we chose $\beta \leqslant 1/10r$ it follows that $e^{-\varepsilon R/(1+\beta)} \leqslant e^{-\varepsilon R + \varepsilon/5}$.

We conclude that $g(v)$ is pointwise within a $\exp(\pm\varepsilon/2)$ factor of the ideal density that gives $\varepsilon/10$-differential privacy by our choice of parameters. Hence, our mechanism satisfies ε-differential privacy. Further since $K \subseteq \widehat{K} \subseteq 2K$, it follows that our mechanism satisfies the stated error bound. Finally, the bound on the moments of the Gamma distribution from Fact 2.10.5 implies that the expected running time of this algorithm is polynomial in N, d, ε^{-1}. ∎

3.7.1 An efficient implementation of NiKM

Theorem 3.7.3 extends to our mechanism for the non-isotropic case.

Theorem 3.7.5. *There is a mechanism M' with runtime polynomial in N, d and ε^{-1} such that*

1. *M' is ε-differentially private,*
2. $\mathrm{err}(M', F) = O(\mathrm{err}(NiKM(F, d, \varepsilon), F))$.

Proof. To implement $\mathrm{NiKM}(F, d, \varepsilon)$ efficiently, we additionally need to compute the subspaces U and V to project onto (Step 2 of the algorithm). Note that these subspaces themselves depend only on the query F and not on the database x. Thus these can be published and the mechanism maintains its privacy for an arbitrary choice of subspaces U and V. The choice of U, V in Section 3.6 depended on the covariance matrix M, which we do not know how to compute exactly. We next describe a method to choose U and V that is efficient such that the resulting mechanism has essentially the same error. The sampling from K can then be replaced by approximate sampling as in the previous subsection, resulting in a polynomial-time differentially private mechanism with small error.

Without loss of generality, K has the property that $B_2^d \subseteq K \subseteq dB_2^d$. In this case, $x_i x_j \leqslant d^2$ so that with $O(d^2 \log d)$ (approximately uniform) samples from K, Chernoff bounds imply that the sample covariance matrix approximates the covariance matrix well in every entry. In other words, we can construct a matrix $\widetilde{M}$ such that with high probability each entry of $\widetilde{M}$ is within $neg(d)$ of the corresponding entry in M. Here and in the rest of the section, $neg(d)$ denotes a function bounded from above by d^{-C} for a large enough constant $C > 0$. The constant varies depending on the context. We also note that with high probability $\widetilde{M}$ is positive semidefinite. This uses the fact that $K \supseteq B_2^d$.

Let the eigenvalues of $\widetilde{M}$ be $\widetilde{\sigma}_1, \ldots, \widetilde{\sigma}_d$ with corresponding eigenvectors $\widetilde{u}_1, \ldots, \widetilde{u}_d$. Let $\widetilde{T} = \widetilde{M}^{-\frac{1}{2}}$, and let $\widetilde{P}$ be the projection operator onto the span of the first d' eigenvectors of $\widetilde{M}$. This defines our subspaces $\widetilde{U}$ and $\widetilde{V}$, and hence the mechanism. We next argue that Lemma 3.6.9 continues to hold.

First note that for any $i \geqslant d' + 1$

$$\mathop{\mathbb{E}}_{a \in K} \langle a, \widetilde{u}_i \rangle^2 = \left| \widetilde{u}_i^T M \widetilde{u}_i \right| = |\widetilde{u}_i^T \widetilde{M} \widetilde{u}_i| + |\widetilde{u}_i^T (M - \widetilde{M}) \widetilde{u}_i| = \widetilde{\sigma}_i + neg(d).$$

Thus, Equation 3.21 continues to hold with $\widetilde{\sigma}_{d'+1}$ replacing $\sigma_{d'+1}$.

To prove that Proposition 3.6.7 continues to hold (with $\widetilde{M}, \widetilde{T}, \widetilde{P}$ replacing M, T, P), we note that the only place in the proof that we used that M is in fact the covariance matrix of K is (3.19), when we require TK to be isotropic. We next argue that (3.19) holds for $\widetilde{T}K$ if $\widetilde{M}$ is a good enough approximation to M. This would imply Proposition 3.6.7 and hence the result.

First recall that Wedin's theorem [Wed] states that for non-singular matrices R, $\widetilde{R}$,

$$\|R^{-1} - \widetilde{R}^{-1}\|_2 \leqslant \frac{1+\sqrt{5}}{2}\|R - \widetilde{R}\|_2 \cdot \max\left\{\|R^{-1}\|_2^2, \|\widetilde{R}^{-1}\|_2^2\right\}.$$

Using this for the matrices $M^{\frac{1}{2}}, \widetilde{M}^{\frac{1}{2}}$ and using standard perturbation bounds gives (see e.g. [KM1]):

$$\|\widetilde{T} - T\|_2 \leqslant O(1) \cdot \|T\|_2^2 \cdot \|\widetilde{M}^{\frac{1}{2}} - M^{\frac{1}{2}}\|_2. \tag{3.23}$$

Since $\|T\|_2$ is at most $poly(d)$ and the second term is $neg(d)$, we conclude that $\|\widetilde{T} - T\|_2$ is $neg(d)$. It follows that

$$TK \subseteq \widetilde{T}K + neg(d)B_2^d. \tag{3.24}$$

Moreover, since TK is in isotropic position, it contains a ball $\frac{1}{d}B_2^d$. It follows from the next lemma (applied to $A = d\widetilde{T}K$) that

$$\frac{1}{2d}B_2^d \subseteq \widetilde{T}K. \tag{3.25}$$

Lemma 3.7.6. *Let A be a convex body in $\mathbb{R}^d$ such that $B_2^d \subseteq A + rB_2^d$ for some $r < 1$. Then a dilation $(1-r)B_2^d$ is contained in A.*

Proof. Let $z \in \mathbb{R}^d$ be a unit vector. Suppose that $z' = (1-r)z \notin A$. Then by the Separating Hyperplane theorem (see, e.g., [BV]), there is a hyperplane H separating z' from A. Thus there is a unit vector w and a scalar b such that $\langle z', w\rangle = b$ and $\langle u, w\rangle < b$ for all $u \in A$. Let $v = z' + rw$. Then by triangle inequality, $\|v\| \leqslant 1$. Moreover,

$$d(v, A) = \inf_{u\in A}\|u - v\| \geqslant \inf_{u\in A}\langle v - u, w\rangle \geqslant b + r - \sup_{u\in A}\langle u, w\rangle > r.$$

This however contradicts the assumption that that $v \in B_2^d \subseteq A + rB_2^d$. Since z was arbitrary, the lemma is proved. ∎

We can thus conclude

$$\begin{aligned}
\left(1 - \tfrac{1}{d}\right)TK &\subseteq \left(1 - \tfrac{1}{d}\right)\widetilde{T}K + neg(d)B_2^d && \text{(using (3.24))}\\
&\subseteq \left(1 - \tfrac{1}{d}\right)\widetilde{T}K + neg(d)\widetilde{T}K && \text{(using (3.25))}\\
&\subseteq \widetilde{T}K,
\end{aligned}$$

where the last containment follows from the fact that $\widetilde{T}K$ is convex and contains the origin. Thus

$$(1-\frac{1}{d})\widetilde{P}TK \subseteq \widetilde{P}\widetilde{T}K. \tag{3.26}$$

Since Corollary 3.2.4 still lower bounds the volume of $\widetilde{P}TK$, we conclude from (3.26) that

$$\mathrm{Vol}_k(\widetilde{P}\widetilde{T}K)^{1/k} \geqslant \frac{1}{e}\mathrm{Vol}_k(\widetilde{P}TK)^{1/k} \geqslant \frac{\alpha_K^{\frac{d-k}{k}}}{e},$$

where we have used the fact that $k \leqslant d$ so that $(1-\frac{1}{d})^k \geqslant \frac{1}{e}$. For $k = d'$, $\frac{d-k}{k}$ is $\Theta(1)$ so that $\mathrm{Vol}_k(\widetilde{P}\widetilde{T}K)^{1/(d-k)} \geqslant \Omega(\alpha_K)$. Thus we have shown that up to constants, (3.19) holds for $\mathrm{Vol}_k(\widetilde{P}\widetilde{T}K)^{1/(d-k)}$ which completes the proof. ■

Chapter 4

A Multiplicative Weights Framework for Interactive Query Release

In this chapter we consider the problem of answering a large number of statistical queries (cf. Section 2.2.1) in a differentially private manner. Our discussion of prior work in Section 2.6 left open the following important questions:

1. Is there a an interactive mechanism that runs in time poly(N) on each of the k queries with non-trivial error on all databases?
2. Could its accuracy (in terms of k, n) match the sampling error $O(\sqrt{\log k/n})$?
3. Is there an interactive mechanism for handling many statistical queries that achieves $(\varepsilon, 0)$-differential privacy (rather than (ε, δ))?
4. Given the cryptographic hardness results for releasing differentially private synthetic data from Section 2.8.1, we cannot hope for sub-linear running time in N (if our algorithm is able to produce synthetic data). Do there exist mechanisms that match or nearly-match this hardness result?
5. What are open avenues for side-stepping the hardness results for synthetic data? Namely, are there meaningful relaxations that permit mechanisms whose running time is sub-linear or even poly-logarithmic in N?

4.1 Main results

Our main contribution is a new privacy-preserving interactive mechanism for answering statistical queries, which we will refer to as the private multiplicative weights (PMW) mechanism. It allows us to give positive answers to the first four questions above, and to make partial progress on the last question. We proceed with a summary of our contributions, see Figure 4.1 for a comparison with the related work. We note that throughout this section,

when we refer to a mechanism's running time as being polynomial or linear, we are measuring the running time as a function of the data universe size N (which may be quite large for high-dimensional data).

A new interactive mechanism. The PMW mechanism runs in time linear in N and provides a worst-case accuracy guarantees for *all* input databases. The mechanism is presented in Figure 4.2, its performance stated in the theorem below. The proof is in Section 5.2. See Section 6.2 for the formal definitions of accuracy and differential privacy for interactive mechanisms.

Theorem 4.1.1. *Let $\mathcal{U}$ be a data universe of size N. For any $k, \varepsilon, \delta, \beta > 0$, the Private Multiplicative Weights Mechanism of Figure 4.2, is an (ε, δ)-differentially private interactive mechanism.*

For any database of size n, the mechanism is (α, β, k)-accurate for (adaptive) statistical queries over $\mathcal{U}$, where

$$\alpha = O\left(\frac{\sqrt{\log(k/\beta)\log(1/\delta)}\log^{1/4} N}{\sqrt{\varepsilon n}}\right)$$

The running time in answering each query is $N \cdot \mathrm{poly}(n) \cdot \mathrm{polylog}(1/\beta, 1/\varepsilon, 1/\delta)$.

The error as a function of n and k grows roughly as $\sqrt{\frac{\log k}{n}}$. Even for blatant non-privacy in the non-interactive setting this dependence on k and n is necessary. See our discussion of lower bounds in Section 2.8. Thus in terms of k and N our upper bound matches a lower bound that holds for a much weaker notion of privacy. In fact, as argued in Section 2.6 $\sqrt{\log k/n}$ is the statistical sampling error observed when computing the maximum error of k insensitive statistics on a sample of size n.

Moreover, the running time is only *linear* in N (for each of the k queries), nearly tight with the cryptographic hardness results of [DNR+]. Previous work even in the non-interactive setting had higher polynomial running time. Finally, we remark that this mechanism can also be used to generate a synthetic database with similar error and running time bounds (in the non-interactive setting), see below for this extensions.

Achieving $(\varepsilon, 0)$-differential privacy. Prior to our work it was conceivable that there was no $(\varepsilon, 0)$-differentially private interactive release mechanism handling, say, n^2 statistical queries with non-trivial error. However, using our multiplicative weights framework, we can achieve the guarantees of Theorem 4.4.1 also with $(\varepsilon, 0)$-differential privacy except for a somewhat worse dependence on n and $\log N$.

Theorem 4.1.2. *Let $\mathscr{U}$ be a data universe of size N. For any $k, \varepsilon, \beta > 0$, there is an $(\varepsilon, 0)$-differentially private interactive mechanism such that: For any database of size n, the mechanism is (α, β, k)-accurate for (adaptive) statistical queries over $\mathscr{U}$, where*

$$\alpha = O\left(\frac{\log(k/\beta)^{1/3} \log^{1/3} N}{(\varepsilon n)^{1/3}}\right).$$

The running time in answering each query is $N \cdot \mathrm{poly}(n) \cdot \mathrm{polylog}(1/\beta, 1/\varepsilon, 1/\delta)$.

We also show a new lower bound on the error that any $(\varepsilon, 0)$-differentially private mechanism must have even in the non-interactive setting when answering $k \gg n$ statistical queries. A novelty of our lower bound is that it simultaneously depends on $n, \log k, \log N$.

Theorem 4.1.3. *Let n be sufficiently large and let $\varepsilon > 0$ be a constant independent of n. Then, for every $k \geqslant n^{1.1}$ there is a set of k statistical queries over a universe of size N such that every $(\varepsilon, 0)$-differentially private mechanism for databases of size n must have error*

$$\alpha \geqslant \Omega(1) \cdot \left(\frac{\log k \cdot \log\left(\frac{N}{n}\right)}{\varepsilon n}\right)^{1/2}$$

with probability 1/2.

Relaxed Notions of Utility. To answer Question 5 that was raised in the introduction, we begin with a discussion of the negative results of [DNR+] and possible avenues for side-stepping them. The negative results for producing synthetic data can be side-stepped by a mechanism whose output has a different format. This is a promising avenue, but synthetic data is a useful output format. It is natural to try to side-step hardness while continuing to output synthetic data. One possibility is working for restricted query classes, but recent work of Ullman and Vadhan [UV] shows hardness even for very simple and natural query classes such as conjunctions. In the known hardness results, however, the *databases* (or rather database distributions) that are hard to sanitize are (arguably) "unnatural", containing cryptographic data in [DNR+] and PCP proofs for the validity of digital signatures in [UV]. Thus, a natural approach to side-stepping hardness is relaxing the utility requirement, and not requiring accuracy for *every* input database.

A mechanism that works only for some input databases is only as interesting as the class of databases for which accuracy is guaranteed. For example, getting accuracy w.h.p. for *most* databases is simple, since (speaking loosely and informally) *most* databases behave like a uniformly random database. Thus, we can get privacy and accuracy by ignoring the input database (which

gives perfect privacy) and answering according a new database drawn uniformly at random (which, for most input databases, will give fairly accurate answers).

Smooth databases and sublinear time. We consider accuracy guarantees for the class of *(pseudo)-smooth* databases. Intuitively, we think of these as databases sampled i.i.d. from *smooth* underlying distributions over the data universe $\mathcal{U}$ of size N. I.e., underlying distributions that do not put too much weight on any particular data item (alternatively, they have high min-entropy). We say that a histogram or distribution y over $\mathcal{U}$ is ξ-*smooth*, if for every $u \in \mathcal{U}$, the probability of u by y is at most ξ. We say that a histogram $x \in \mathbb{R}_+^N$ is (ξ, ϕ)-*pseudo-smooth w.r.t a set $\mathcal{Q}$ of queries* if there exists some ξ-smooth y that approximates it well w.r.t every query in $\mathcal{Q}$. I.e., for every $f \in \mathcal{Q}$, $|f(y) - f(x)| \leqslant \phi$ (where by $f(y)$ we mean the expectation of f over data items drawn from y). See Section 4.6 for formal definitions.

The PMW mechanism yields a mechanism with improved running time—sub-linear, or even polylogarithmic in N—for pseudo-smooth databases. The new mechanism (with smoothness parameter ξ) runs in time that depends linearly on ξN rather than N. It guarantees differential privacy for *any* input database. Its error is similar to that of the mechanism of Theorem 4.4.1 (up to an additional ϕ error), but this accuracy guarantee is only: (i) for a set $\mathcal{Q}$ of interactive statistical queries that are fixed in advance (i.e. non-adaptively). We note that the mechanism is interactive in the sense that it need not know the queries in advance, but accuracy is not guaranteed for adversarially chosen queries (see the discussion in Section 4.2 for motivation for this relaxation), and (ii) for input databases that are (ξ, ϕ)-smooth with respect to the query class $\mathcal{Q}$. The performance guarantees are in Theorem 4.1.4 below. The proof is in Section 4.6

Theorem 4.1.4. *Let $\mathcal{U}$ be a data universe of size N. For any $\varepsilon, \delta, \beta, \xi, \phi > 0$, the Private Multiplicative Weights Mechanism of Figure 4.2 is an (ε, δ)-differentially private interactive mechanism. Moreover, for any sequence $\mathcal{Q}$ of k interactive statistical queries over $\mathcal{U}$ that are fixed in advance (non-adaptively), for any database of size n that is (ξ, ϕ)-pseudo-smooth w.r.t $\mathcal{Q}$, the mechanism is (α, β, k)-non adaptively accurate w.r.t. $\mathcal{Q}$, where*

$$\alpha = \widetilde{O}\left(\phi + \frac{\log(1/\delta)\log^{1/4}(\xi N) \cdot (\log k + \log(1/\beta))}{\sqrt{n} \cdot \varepsilon}\right).$$

The running time in answering each query is $(\xi N) \cdot \mathrm{poly}(n) \cdot \mathrm{polylog}(1/\beta, 1/\varepsilon, 1/\delta, 1/\xi, 1/\phi)$.

In particular, for very good smoothness $\xi = \mathrm{polylog} N/N$, the running time will depend only poly-logarithmically on N. The main observation for

achieving this improved running time is that for (pseudo)-smooth databases we can effectively reduce the data universe size by sub-sampling, and then apply our algorithm to the smaller data universe. The mechanism does not require knowledge of the histogram which certifies that the given input database is pseudo-smooth.

The privacy guarantee is the standard notion of differential privacy. I.e., privacy holds always and for every database. The accuracy guarantee is only for pseudo-smooth databases, and we interpret it as follows. The data set is drawn i.i.d from an unknown underlying distribution D (the standard view in statistics). The mechanism guarantees accuracy and sub-linear efficiency as long as the underlying data distribution is smooth. If the underlying distribution is ξ-smooth, then w.h.p. the database x (which we think of as being drawn i.i.d from D and of large enough size) is "close" to D on every query, and so w.h.p. x is (ξ, ϕ)-smooth and the mechanism is accurate. An important aspect of this guarantee is that *there is no need to know what the underlying distribution is*, only that it is smooth. A promising approach in practice may be to run this mechanism as a very efficient heuristic. The heuristic *guarantees* privacy, and also has a rigorous accuracy guarantee under assumptions about the underlying distribution. We note that Dwork and Lei [DL] also proposed mechanisms that always guarantee privacy, but guarantee accuracy only for a subset of databases (or underlying distributions).

We also note that [RR] considered databases drawn from a distribution that was itself picked randomly from the set of all distributions. Such "random distributions" are indeed very smooth (w.h.p. $\xi \leqslant O(\log N/N)$) and therefore a special case of our model.

Figure 4.1 summarizes and compares relevant past work on answering statistical queries. We proceed with an overview of techniques.

4.2 Overview of proof and techniques

Multiplicative Weights. We use a (privacy-preserving) multiplicative weights mechanism (see [LW, AHK]). The mechanism views databases as histograms or distributions (also known as "fractional" databases) over the data universe $\mathcal{U}$ (as was done in [DNR+]). At a high level, the mechanism works as follows. The real database being analyzed is x. Here, we view x as distribution or normalized histogram over $\mathcal{U}$, with positive weight on the data items in x. See Section 2.1.1 for more information on histograms. We will use the words histogram and database interchangeably throughout this chapter. Since we work with normalized histograms we can think of a statistical query simple as a vector $f \in [0,1]^N$ with $f(x) = \langle f, x \rangle$.

Mechanism	**interactive?**	**error** in terms of n, N, k	**runtime**	privacy	remark
[DMNS]	√	$\sqrt{k}$	—	(ε, δ)	
[BLR]	—	$n^{2/3}\log^{1/3} N \cdot \log^{1/3} k$	$N^{O(n)}$	$(\varepsilon, 0)$	
[DNR+]	—	$\sqrt{n \log N} \cdot k^{o(1)}$	$\text{poly}(N)$	(ε, δ)	
[DRV]	—	$\sqrt{n \log N} \cdot \log^2 k$	$\text{poly}(N)$	(ε, δ)	
[RR]	√	$n^{2/3}\log^{1/3} N \cdot \log k$	$N^{O(n)}$	(ε, δ)	
[RR]	√	$n^{2/3}\log^{1/3} N \cdot \log k$	$\text{poly}(N)$	(ε, δ)	random DBs
This work	√	$\sqrt{n \log k}\log^{1/4} N$	$\widetilde{O}(N)$	(ε, δ)	
This work	√	$n^{2/3}\log^{1/3} N \log^{1/3} k$	$\widetilde{O}(N)$	$(\varepsilon, 0)$	
This work	√	$\sqrt{n \log k}\log\log N \cdot \log k$	$\text{polylog}N$	(ε, δ)	smooth DBs

Figure 4.1: Comparison to previous work for k linear queries each of sensitivity 1. For simplicity the dependence on δ is omitted from the comparison. Runtime stated in terms of N omitting other factors. Error bounds are a factor n larger than throughout the chapter and accurate up to polylogarithmic factors. Note that random databases are a special case of smooth databases (see Section 4.6).

The mechanism also maintains an updated histogram, denoted as x_t at the end of round t. In each round t, after the t-th statistical query f_t has been specified, x_{t-1} is updated to obtain x_t. The initial database x_0 is simply the uniform distribution over the data universe. I.e., each coordinate $u \in \mathcal{U}$ has weight $1/N$.

In the t-th round, after the t-th query f_t has been specified, we compute a noisy answer $\widehat{a_t}$ by adding (properly scaled) Laplace noise to $f_t(x)$—the "true" answer on the real database. We then compare this noisy answer with the answer given by the previous round's database $f_t(x_{t-1})$. If the answers are "close", then this is a "lazy" round, and we simply output $f_t(x_{t-1})$ and set $x_t \leftarrow x_{t-1}$. If the answers are "far", then this is an "update" round and we need to update or "improve" x_t using a multiplicative weights re-weighting. The intuition is that the re-weighting brings x_t "closer" to an accurate answer on f_t. In a nutshell, this is all the algorithm does. The only additional step required is bounding the number of "update" rounds: if the total number of update rounds grows to be larger than (roughly) n, then the mechanism fails and terminates. This will be a low probability event. See Figure 4.2 for the details. Given this overview of the algorithm, it remains to specify how to: (i) compute $f_t(x_{t-1})$, and (ii) re-weight or improve the database on update rounds. We proceed with an overview of the arguments for accuracy and privacy.

For this exposition, we think of the mechanism as explicitly maintaining the x_t databases, resulting in complexity that is roughly linear in $N = |\mathcal{U}|$.

Using standard techniques we can make the memory used by the mechanism logarithmic in N (computing each coordinate of x_t as it is needed). Either way, it is possible to compute $f_t(x_{t-1})$ in linear time.

The re-weighting (done only in update rounds), proceeds as follows. If in the comparison we made, the answer according to x_{t-1} was "too small", then we increase by a small multiplicative factor the weight of items $u \in \mathcal{U}$ that satisfy the query f_t's predicate, and decrease the weight of those that do not satisfy it by the same factor. If the answer was "too large" then do the reverse in terms of increasing and decreasing the weights. We then normalize the resulting weights to obtain a new database whose entries sum to 1. The intuition, again, is that we are bringing x_t "closer" to an accurate answer on f_t. The computational work scales linearly with N.

To argue accuracy, observe that as long as the number of update rounds stays below the (roughly n) threshold, our algorithm ensures bounded error (assuming the Laplace noise we add is not too large). The question is whether the number of update rounds remains small enough. This is in fact the case, and the proof is via a multiplicative weights potential argument. Viewing databases as distributions over $\mathcal{U}$, we take the *potential* of database y to be the relative entropy $\mathrm{RE}(x\|y)$ between y and the real database x. We show that if the error of x_{t-1} on query f_t is large (roughly larger than $1/\sqrt{n}$), then the potential of the re-weighted x_t is smaller by at least (roughly) $1/n$ than the potential of x_{t-1}. Thus, in every "update" round, the potential drops, and the drop is significant. By bounding the potential of x_0, we get that the number of update rounds is at most (roughly) n.

"Pay as you go" privacy analysis. At first glance, privacy might seem problematic: we access the database and compute a noisy answer *in every round*. Since the number of queries we want to answer (number of rounds) might be huge, unless we add a huge amount of noise this collection of noisy answers *is not privacy preserving*. The point, however, is that in most rounds we don't release the noisy answer. All we do is check whether or not our current database x_{t-1} is accurate, and if so we use it to generate the mechanism's output. In all but the few update rounds, the perturbed true answer is not released, and we want to argue that privacy in all those lazy rounds comes (essentially) "for free". The argument builds on ideas from privacy analyses in previous works [DNR+, DNPR, RR]).

A central concern is arguing that the "locations" of the update rounds be privacy-preserving (there is an additional, more standard, concern that the noisy answers in the few update rounds also preserve privacy). Speaking intuitively (and somewhat inaccurately), for any two adjacent databases, there are w.h.p. only roughly n "borderline" rounds, where the noise is such that

on one database this round is update and on another this round is lazy. This is because, conditioning on a round being "borderline", with constant probability it is actually an "update" round. Since the number of update rounds is at most roughly n, with overwhelming probability the number of borderline rounds also is roughly n. For non-borderline rounds, those rounds' being an update or a lazy round is determined similarly for the two databases, and so privacy for these rounds come "for free". The borderline rounds are few, and so the total privacy hit incurred for them is small.

Given this intuition, we want to argue that the "privacy loss", or "confidence gain" of an adversary, is small. At a high level, if we bound the worst-case confidence gain in each update round by roughly $O(\varepsilon/\sqrt{n})$, then by an "evolution of confidence" argument due to [DN1, DN2, DRV], the total confidence gain of an adversary over the roughly n update rounds will be only ε w.h.p. To bound the confidence gain, we define "borderline" rounds as an event over the noise values on a database x, and show that: (1) Conditioned on a round being borderline on x, it will be an update round on x w.h.p. This means borderline rounds are few. (2) Conditioned on a round being borderline on x, the worst-case confidence gain of an adversary viewing the mechanism's behavior in this round on x vs. an adjacent x' is bounded by roughly $\varepsilon/\sqrt{n}$. This means the privacy hit in borderline rounds isn't too large, and we can "afford" roughly n of them. (3) Conditioned on a round *not* being borderline, there is no privacy loss in this round on x vs. any adjacent x'. I.e., non-borderline rounds come for free (in terms of privacy).

This analysis allows us to add less noise than previous works, while still maintaining (ε, δ) differential privacy. It may find other applications in interactive or adaptive privacy settings. Details are in Section 4.3.2.

Sublinear Time Mechanism for Smooth Databases. We observe that we can modify the PMW mechanism to work over a smaller data universe $V \subseteq \mathcal{U}$, as long as there *exists* a database x^* whose support is only over V, and gives close answers to those of x on every query we will be asked. We modify the algorithm to maintain multiplicative weights only over the smaller set V, and increase slightly the inaccuracy threshold for declaring a round as "update". For the analysis, we modify the potential function: it measures relative entropy to x^* rather than x. In update rounds, the distance between x_{t-1} and this new x^* on the current query is large (since x^* is close to x, and x_{t-1} is far from x). This means that re-weighting will reduce $\mathrm{RE}(x^* \| x_{t-1})$, and even though we maintain multiplicative weights only over a smaller set V, the number of update rounds will be small. Maintaining multiplicative weights over V rather than $\mathcal{U}$ reduces the complexity from linear in $|\mathcal{U}|$ to linear in $|V|$.

To use the above observation, we argue that for any large set of statistical queries $\mathcal{Q}$ and any (ξ, ϕ)-pseudo-smooth database x, if we choose a uniformly random small (but not too small) sub-universe $V \subseteq \mathcal{U}$, then w.h.p there *exists* x^* whose support is in V that is close to x on all queries in $\mathcal{Q}$. In fact, sampling a sub-universe of size roughly $\xi N \cdot n \cdot \log|\mathcal{Q}|$ suffices. This means that indeed PMW can be run on the reduced data universe V with reduced computational complexity. See Section 4.6.1 for this argument.

Utility here is for a fixed non-adaptive set $\mathcal{Q}$ of queries (that need not be known in advance). We find this utility guarantee to still be well motivated—note that, privacy aside, the input database itself, which is sampled i.i.d from an underlying distribution, isn't guaranteed to yield good answers for adaptively chosen queries). Finally, we remark that this technique for reducing the data universe size (the data dimensionality) may be more general than the application to PMW. In particular, previous mechanisms such as [DNR+, DRV] can also be modified to take advantage of this sampling and obtain improved running time for smooth databases (the running time will be polynomial, rather than linear as it is for the PMW mechanism).

Synthetic databases. We conclude by noting that the PMW mechanism can be used to generate synthetic data (in the non-interactive setting). To do this, iterate the mechanism over a set of queries $\mathcal{Q}$, repeatedly processing all the queries in $\mathcal{Q}$ and halting when either (i) we made roughly $n+1$ iterations, i.e. have processed every query in $\mathcal{Q}$ n times, or (ii) we have made a complete pass over all the queries in $\mathcal{Q}$ without any update rounds (whichever of these two conditions occurs first). If we make a complete pass over $\mathcal{Q}$ without any update rounds, then we know that the x_t we have is accurate for all the queries in $\mathcal{Q}$ and we can release it (or a subsample from it) as a privacy-preserving synthetic database. By the potential argument, there can be at most roughly n update rounds. Thus, after $n+1$ iterations we are guaranteed to have a pass without any update rounds. Previous mechanisms for generating synthetic databases involved linear programming and were more expensive computationally.

4.2.1 Preliminaries

In this section we review some preliminaries specific to the interactive setting.

Accuracy and privacy in the interactive setting. Formally, an *interactive mechanism* $M(x)$ is a stateful randomized algorithm which holds a histogram $x \in \mathbb{R}^N$. It receives successive statistical queries $f_1, f_2, \ldots \in \mathcal{F}$ one by one, and in each round t, on query f_t, it outputs a (randomized) answer a_t (a function of the input histogram, the internal state, and the mechanism's coins).

For privacy guarantees, we *always* assume that the queries are given to the mechanism in an adversarial and adaptive fashion by a randomized algorithm A called the *adversary*. For accuracy guarantees, while we usually consider adaptive adversarial, we will also consider non-adaptive adversarial queries chosen in advance—we still consider such a mechanism to be interactive, because it does not know in advance what these queries will be. The main query class we consider throughout this work is the class $\mathcal{F}$ of all statistical queries, as well as sub-classes of it.

Definition 4.2.1 ((α,β,k)-Accuracy in the Interactive Setting)**.** We say that a mechanism M is (α,β,k)-(adaptively) accurate for a database x, if when it is run for k rounds, for any (adaptively chosen) statistical queries, with all but β probability over the mechanism's coins $\forall t \in [k], |a_t - f_t(x)| \leqslant \alpha$.

We say that a mechanism M is (α,β,k)-non adaptively accurate for a query sequence $\mathcal{Q}$ of size k and a database x, if when it is run for k rounds on the queries in $\mathcal{Q}$, with all but β probability over the mechanism's coins $\forall t \in [k], |a_t - f_t(x)| \leqslant \alpha$.

For privacy, the interaction of a mechanism $M(x)$ and an adversary A specifies a probability distribution $[M(x), A]$ over *transcripts*, i.e., sequences of queries and answers $(f_1, a_1, f_2, a_2, \ldots, f_k, a_k)$. Let $\mathrm{Trans}(\mathcal{F}, k)$ denote the set of all transcripts of any length k with queries from $\mathcal{F}$. We will assume that the parameter k is known to the mechanism ahead of time. Our privacy requirement asks that the entire transcript satisfies differential privacy.

Definition 4.2.2 ((ε,δ)-Differential Privacy in the Interactive Setting)**.** We say a mechanism M provides (ε,δ)-differential privacy for a class of queries $\mathcal{F}$, if for every adversary A and every two histograms $x, x' \in \mathbb{R}^N$ satisfying $\|x - x\|_1 \leqslant 1/n$, the following is true: Let $P = [M(x), A]$ denote the transcript between $M(x)$ and A. Let $Q = [M(x'), A]$ denote the transcript between $M(x')$ and A. Then, for every $S \subseteq \mathrm{Trans}(\mathcal{F}, k)$, we have

$$P(S) \leqslant e^{\varepsilon} Q(S) + \delta.$$

4.3 Private multiplicative weights mechanism

In the PMW mechanism of Figure 4.2, in each round t, we are given a linear query f_t over $\mathscr{U}$ and x_t denotes a fractional histogram (distribution over $V \subseteq \mathscr{U}$) computed in round t. The domain of this histogram is V rather than $\mathscr{U}$. Here, V could be much smaller than $\mathscr{U}$ and this allows for some flexibility later, in proving Theorem 4.1.4, where we aim for improved efficiency. For this section, unless otherwise specified, we assume that $V = \mathscr{U}$. In particular

Parameters: A subset of the coordinates $V \subseteq \mathscr{U}$ with $|V| = M$ (by default $V = \mathscr{U}$), intended number of rounds $k \in \mathbb{N}$, privacy parameters $\varepsilon, \delta > 0$ and failure probability $\beta > 0$. See (4.2) for the setting of η, σ, T.
Input: Database $D \in \mathscr{D}_n$ corresponding to a histogram $x \in \mathbb{R}^N$
Algorithm: Set $y_0[u] = x_0[u] = 1/M$ for all $u \in V$
In each round $t \leftarrow 1, 2 \ldots, k$ when receiving a linear query f_t do the following:

1. Sample $A_t \sim Lap(\sigma)$. Compute the noisy answer $\widehat{a_t} \leftarrow f_t(x) + A_t$.

2. Compute the difference $\widehat{d_t} \leftarrow f_t(x_{t-1}) - \widehat{a_t}$:

   If $|\widehat{d_t}| \leqslant T$, then set $w_t \leftarrow 0$, $x_t \leftarrow x_{t-1}$, output $f_t(x_{t-1})$, and proceed to the next iteration.

   If $|\widehat{d_t}| > T$, then set $w_t \leftarrow 1$ and:

   — for all $u \in V$, update

   $$y_t[u] \leftarrow x_{t-1}[u] \cdot \exp(-\eta \cdot r_t[u]), \tag{4.1}$$

   where $r_t[u] = f_t[u]$ if $\widehat{d_t} > 0$ and $r_t[u] = 1 - f_t[u]$ otherwise.

   — Normalize, $x_t[u] \leftarrow \frac{y_t[u]}{\sum_{u \in V} y_t[u]}$.

   — If $\sum_{j=1}^{t} w_j > \eta^{-2} \log M$, then abort and output "failure". Otherwise, output the noisy answer $\widehat{a_t}$ and proceed to the next iteration.

Figure 4.2: Private Multiplicative Weights (PMW) Mechanism

this is the case in the statement of Theorem 4.4.1, the main theorem that we prove in this section.

We use a_t to denote the true answer on the database on query t, and $\widehat{a_t}$ denotes this same answer with noise added to it. We use d_t to denote the difference between the true answer a_t and the answer given by x_{t-1}, i.e.,

$$d_t = f_t(x_{t-1}) - f_t(x).$$

We denote by $\widehat{d_t}$ the difference between the *noisy* answer and the answer given by x_{t-1}. The boolean variable w_t denotes whether the noisy difference was large or small. If $\widehat{d_t}$ is smaller (in absolute value) than $\approx 1/\sqrt{n}$, then this round is *lazy* and we set $w_t = 0$. If $\widehat{d_t}$ is larger than threshold then this is an *update* round and we set $w_t = 1$.

Choice of parameters. We set the parameters η, σ, T as follows:

$$\eta = \sqrt{\frac{\log^{1/2} M \log(k/\beta)\log(1/\delta)}{\varepsilon n}} \qquad \sigma = \frac{10\eta}{\log(k/\beta)} \qquad T = 40\eta. \tag{4.2}$$

To understand the choice of parameters, let $m = \eta^{-2}\log M$ denote the bound on the number of update rounds ensured by our algorithm. We chose our parameters in (4.2) such that the following two relations hold

$$\sigma n \geqslant \frac{10\sqrt{m}\log(1/\delta)}{\varepsilon} \qquad \text{and} \qquad T \geqslant 4\sigma \log(k/\beta). \tag{4.3}$$

Intuitively speaking, the first condition ensures that the scaling σ of the Laplacian variables used in our algorithm is large enough to handle m update rounds while providing (ε, δ)-differential privacy. The second condition ensures that the Laplacian variables are small compared to the threshold T. Subject to these two constraints expressed in (4.3), our goal is to minimize η and σ. This is because η controls how large the noise magnitude σ has to be which in turns determines the threshold T. The error of our algorithm must scale with T.

We are now ready to prove Theorem 4.4.1, i.e. the utility and privacy of the PMW mechanism. This follows directly from our utility analysis provided in Section 5.2.1 and our privacy argument presented in Section 4.3.2.

4.3.1 Utility analysis

To argue utility, we need to show that even for very large total number of rounds k, the number of update rounds is at most roughly n with high probability. This is done using a potential argument. Intuitively, the potential of a database x_t is the relative entropy between the true histogram x and our estimate x_t.

Since in general $V \neq \mathscr{U}$, we will actually define the potential with respect to a target histogram $x^* \in \mathbb{R}^N$ with support only over V. This x^* need not be equal to x, nor does it have to be known by the algorithm. This added bit of generality will be useful for us later in Section 4.6 when we modify the mechanism to run in sublinear time. For this section, however, unless we explicitly note otherwise the reader may think of x^* as being equal to x. The potential function is then defined as

$$\Psi_t = \mathrm{RE}(x^* \| x_t) = \sum_{u \in V} x^*[u] \log\left(\frac{x^*[u]}{x_t[u]}\right). \tag{4.4}$$

Note that x^* and x_t are both normalized so that we can think of them both as distributions or histograms over $\mathscr{U}$. We start with two simple observations:

Lemma 4.3.1. $\Psi_0 \leqslant \log M$.

Proof. Indeed, by the nonnegativity of entropy $\mathrm{H}(x^*)$ we get that $\Psi_0 = \log M - \mathrm{H}(x^*) \leqslant \log M$. ∎

Lemma 4.3.2. *For every t, we have $\Psi_t \geqslant 0$.*

Proof. By the nonnegativity of relative entropy (Fact 2.10.1). ∎

Our goal is to show that if a round is an update round (and $w_t = 1$), then the potential drop in that round is at least η^2. In Lemma 4.3.5 we show that this is indeed the case in every round, except with β/k probability over the algorithm's coins. Taking a union bound, we conclude that with all but β probability over the algorithm's coins, there are at most $\eta^{-2}\log M$ update rounds. The next lemma quantifies the potential drop in terms of the penalty vector r_t and the parameter η using a multiplicative weights argument.

Lemma 4.3.3. *In each update round t, we have $\Psi_{t-1} - \Psi_t \geqslant \eta\langle r_t, x_{t-1} - x^*\rangle - \eta^2$.*

Proof. We can rewrite the potential drop as follows:

$$\begin{aligned}
\Psi_{t-1} - \Psi_t &= \sum_{u\in V} x^*[u]\left(\log\left(\frac{x^*[u]}{x_{t-1}[u]}\right) - \log\left(\frac{x^*[u]}{x_t[u]}\right)\right)\\
&= \sum_{u\in V} x^*[u]\log\left(\frac{x_t[u]}{x_{t-1}[u]}\right)\\
&= \sum_{u\in V} x^*[u]\log\left(\exp(-\eta r_t[u])\frac{x_{t-1}[u]}{x_{t-1}[u]\sum_{u\in V} y_t[u]}\right)\\
&= -\eta\langle r_t, x^*\rangle - \sum_{u\in V} x^*[u]\log\left(\sum_{u\in V} y_t[u]\right)\\
&= -\eta\langle r_t, x^*\rangle - \log\left(\sum_{u\in V}\exp(-\eta r_t[u])x_{t-1}[u]\right) \qquad (\text{since } \textstyle\sum x^*[u] = 1)
\end{aligned}$$

Note that

$$\exp(-\eta r_t[u]) \leqslant 1 - \eta r_t[u] + \eta^2 r_t[u]^2 \leqslant 1 - \eta r_t[u] + \eta^2.$$

Using this and $\sum x_{t-1}[u] = 1$ we get

$$\log\left(\sum_{u\in V}\exp(-\eta r_t[u])x_{t-1}[u]\right) \leqslant \log\left(1 - \eta\langle r_t x_{t-1}\rangle + \eta^2\right) \leqslant -\eta\langle r_t, x_{t-1}\rangle + \eta^2,$$

where we used $\log(1+y) \leqslant y$ for $y > -1$. We conclude that

$$\Psi_{t-1} - \Psi_t \geqslant -\eta\langle r_t, x^*\rangle + \eta\langle r_t, x_{t-1}\rangle - \eta^2 = \eta\langle r_t, x_{t-1} - x^*\rangle - \eta^2.$$ ∎

In the following lemmata, we condition on the event that $|A_t| \leqslant T/2$. Since A_t is a centered Laplacian with standard deviation σ and $T \geqslant 4\sigma(\log k + \log(1/\beta))$, this event occurs with all but β/k probability in every round t.

The next lemma connects the inner product $\langle r_t, x^* - x_{t-1}\rangle$ with the "error" of x_{t-1} on the query f_t. Here, error is measured with respect to the true histogram x. To relate x with x^*, we further denote

$$\mathrm{err}(x^*, f_t) = |f_t(x^*) - f_t(x)|. \tag{4.5}$$

When $x^* = x$ we get that $\mathrm{err}(x^*, f_t) = 0$ always, and in general we will be interested in x^* databases where $\mathrm{err}(x^*, f_t)$ is small for all $t \in [k]$.

Lemma 4.3.4. *In each round t where $|\widehat{d_t}| \geqslant T$ and $|A_t| \leqslant T/2$ we have*

$$\langle r_t, x_{t-1} - x^*\rangle \geqslant |f_t(x) - f_t(x_{t-1})| - \mathrm{err}(x^*, f_t).$$

Proof. By assumption $|\widehat{d_t}| \geqslant T$ and $|d_t - \widehat{d_t}| \leqslant |A_t| \leqslant T/2$. Hence, $sign(d_t) = sign(\widehat{d_t})$. We distinguish the two cases where $sign(d_t) < 0$ and $sign(d_t) \geqslant 0$. First, suppose

$$0 > sign(d_t) = sign(f_t(x_{t-1}) - f_t(x)).$$

It follows that $r_t[u] = 1 - f_t[u]$. Hence,

$$\begin{aligned}
\sum_{u\in V} r_t[u](x_{t-1}[u] - x^*[u]) &= -\Big(f_t(x_{t-1}) - f_t(x^*)\Big) + \sum_{u\in V} x^*[u] - \sum_{u\in V} x_{t-1}[u] \\
&= -\Big(f_t(x_{t-1}) - f_t(x^*)\Big) \ \ (\text{using } \textstyle\sum_i x_{t-1}[u] = \sum_i x[u] = 1) \\
&\geqslant -\Big(f_t(x_{t-1}) - f_t(x)\Big) - \mathrm{err}(x^*, f_t) \\
&= |f_t(x_{t-1}) - f_t(x)| - \mathrm{err}(x^*, f_t).
\end{aligned}$$

The case where $sign(d_t) = sign(\widehat{d_t}) \geqslant 0$ is analogous. The claim follows. ∎

Lemma 4.3.5. *In each round t where $|\widehat{d_t}| \geqslant T$ and $A_t \leqslant T/2$ we have*

$$\Psi_{t-1} - \Psi_t \geqslant \eta\left(\frac{T}{2} - \mathrm{err}(x^*, f_t)\right) - \eta^2. \tag{4.6}$$

Proof. By assumption and Lemma 4.3.4, we have $\langle r_t, x^* - x_{t-1}\rangle \geqslant |f_t(x_{t-1}) - f_t(x)| - \mathrm{err}(x^*, f_t)$. We then get from Lemma 4.3.3,

$$\begin{aligned}
\Psi_{t-1} - \Psi_t &\geqslant \eta\langle r_t, x^* - x_{t-1}\rangle - \eta^2 \\
&\geqslant \eta|f_t(x_{t-1}) - f_t(x)| - \eta^2 - \eta\cdot\mathrm{err}(x^*, f_t) \\
&= \eta|d_t| - \eta^2 - \eta\cdot\mathrm{err}(x^*, f_t).
\end{aligned}$$

On the other hand, since $|\widehat{d_t}| \geqslant T$ and $A_t \leqslant T/2$, we have that $|d_t| \geqslant T/2$. This proves the claim. ∎

We are now ready to prove our main lemma about utility.

Lemma 4.3.6 (Utility for $V = \mathscr{U}$). *When the PMW mechanism is run with $V = \mathscr{U}$, it is an (α, β, k)-accurate interactive mechanism, where*

$$\alpha = O\left(\frac{\sqrt{\log(k/\beta)\log(1/\delta)}\log^{1/4} N}{\sqrt{\varepsilon n}}\right)$$

Proof. For $V = \mathscr{U}$, we may choose $x^* = x$ so that $\mathrm{err}(f_t) = 0$ for all $t \in [k]$. Furthermore, with all but β probability over the algorithm's coins, the event $A_t \leqslant T/2$ occurs for every round $t \in [k]$. Hence, by Lemma 4.3.5 and $T \geqslant 4\eta$, the potential drop in every update round is at least

$$\Psi_{t-1} - \Psi_t \geqslant \eta\frac{T}{2} - \eta^2 \geqslant \eta^2.$$

Since $\Psi_0 \leqslant \log N$, the number of update rounds is bounded by $\eta^{-2}\log N$. Hence, by our termination criterion, the algorithm terminates after having answered all k queries. Furthermore, the error of the algorithm is never larger than

$$T + |A_t| \leqslant 2T = O\left(\frac{\sqrt{\log(k/\beta)\log(1/\delta)}\log^{1/4} N}{\sqrt{\varepsilon n}}\right). \qquad \blacksquare$$

We now give a utility analysis in the general case where we are working with a smaller universe $V \subseteq \mathscr{U}$. This will be used (in Section 4.6) to prove the utility guarantee of Theorem 4.1.4. The proof is analogous to the previous one except for minor modifications.

Lemma 4.3.7 (Utility when $V \subsetneq \mathscr{U}$). *Let $f_1, f_2, \ldots, f_k$ denote a sequence of k linear queries. Take*

$$\gamma = \inf_{x^*} \sup_{t \in [k]} \mathrm{err}(x^*, f_t)$$

where x^* ranges over all histograms supported on V. When the PMW mechanism is run with V on the query sequence above, and with threshold parameter $T' = T + \gamma$, it is an (α, β, k)-non adaptively accurate interactive mechanism, where

$$\alpha = O\left(\gamma + \frac{\sqrt{\log(k/\beta)\log(1/\delta)}\log^{1/4} N}{\sqrt{\varepsilon n}}\right).$$

Proof. To prove the lemma, we choose x^* as a minimizer in the definition of γ. With this choice, Lemma 4.3.5 implies that

$$\Psi_{t-1} - \Psi_t \geqslant \eta\left(\frac{T'}{2} - \gamma\right) - \eta^2 \geqslant \eta^2,$$

since we chose $T' \geqslant 4\eta + \gamma$. The argument is now the same as before. In particular, the error is bounded by $O(T') = O(\gamma + T)$ which is what we claimed. $\blacksquare$

4.3.2 Privacy analysis

Our goal in this section is to demonstrate that the interactive mechanism satisfies (ε, δ)-differential privacy (see Definition 4.2.2). We assume that all parameters such as V, σ, η, and T are publicly known. They pose no privacy threat as they do not depend on the input database. For ease of notation we will assume that $V = \mathscr{U}$ throughout this section. The proof is the same for $V \subseteq \mathscr{U}$ (the sub-universe V is always public information).

Simplifying the transcript. Without loss of generality, we can simplify the output of our mechanism (and hence the transcript between adversary and mechanism). We claim that the output transcript of the mechanism is determined by the following (random) vector $\mathbf{v}$. In particular, it is sufficient to argue that $\mathbf{v}$ is differentially private. For every round t, the t-th entry in $\mathbf{v}$ is defined as

$$\mathbf{v}_t = \begin{cases} \perp & \text{if } w_t = 0 \\ \widehat{a_t} & \text{if } w_t = 1 \end{cases}.$$

In other words, $\mathbf{v}_t$ is equal to $\perp$ if that round was a lazy round, or the noisy answer $\widehat{a_t} = f_t(x) + A_t$ if round t was an update round. This is sufficient information for reconstructing the algorithm's output: given the prefix $\mathbf{v}_{<t} = (\mathbf{v}_1, \ldots, \mathbf{v}_{t-1})$, we can compute the current histogram x_{t-1} for the beginning of round t. For the lazy rounds, this is sufficient information for generating the algorithm's output. For the update rounds, $\mathbf{v}_t = \widehat{a_t}$, which is the output for round t. It is also sufficient information for re-weighting and computing the new x_t.

Note that to argue differential privacy, we need to prove that the entire transcript, including the queries of the adversary, is differentially private. Without loss of generality, we may assume that the adversary is deterministic.[1] In this case f_t is determined by $\mathbf{v}_{<t}$. Hence, there is no need to include f_t explicitly in our transcript. It suffices to show that the vector $\mathbf{v}$ is (ε, δ)-differentially private.

Lemma 4.3.8 (Privacy). *The PMW mechanism satisfies (ε, δ)-differential privacy.*

Proof. Fix an adversary and histograms $x, x' \in \mathbb{R}^N$ so that $\|x - x'\|_1 \leqslant 1/n$. Let $\varepsilon_0 = 1/\sigma n$ (where σ is the scaling of the Laplacian variables used in our algorithm).

Let P denote the output distribution of our mechanism when run on the input database x and similarly let Q denote the output of our mechanism

[1] We can think of a randomized adversary as a collection of deterministic adversaries one for each fixing of the adversary's randomness (which is independent of our algorithm's coin tosses).

when run on x'. Both distributions are supported on $\mathcal{S} = (\{\perp\} \cup \mathbb{R})^k$. For $v \in \mathcal{S}$, we define the *loss* function $L \colon \mathcal{S} \to \mathbb{R}$ as

$$L(v) := \log\left(\frac{P(v)}{Q(v)}\right). \tag{4.7}$$

Here and in the following we identify P with its probability density function $\mathrm{d}P$ (which exists by the Radon-Nikodym theorem). Henceforth $P(v)$ denotes the density of P at v.

We will then show that

$$\mathop{\mathbb{P}}_{\mathbf{v}\sim P}\{L(\mathbf{v}) \leqslant \varepsilon\} \geqslant 1 - \delta. \tag{4.8}$$

By Lemma 2.1.3, inequality (4.8) implies (ε, δ)-differential privacy and hence our claim.

Fix a transcript $v \in \mathcal{S}$ we will now proceed to analyze $L(v)$. Using the chain rule for conditional probabilities, let us rewrite $L(v)$ as

$$L(v) = \log\left(\frac{P(v)}{Q(v)}\right) = \sum_{t\in[k]} \log\left(\frac{P_t(v_t \mid v_{<t})}{Q_t(v_t \mid v_{<t})}\right), \tag{4.9}$$

where $P_t(v_t \mid v_{<t})$ denotes the conditional probability (or rather conditional density) of outputting v_t in step t on input histogram x, conditioned on $v_{<t} = (v_1, \ldots, v_{t-1})$. The definition of $Q_t(v_t \mid v_{<t})$ is analogous with x' replacing x. Note that conditioning on $v_{<t}$ is necessary, since the coordinates of the transcript are not independent. Further, it is important to note that conditioned on $v_{<t}$, the estimate x_{t-1} in the algorithm at step t is the same regardless of whether we started from x or x'.

Borderline event. We define an event $S_t = S(v_{<t}) \subseteq \mathbb{R}$ on the noise values as follows. Let $d_t = f_t(x_{t-1}) - f_t(x)$. Note that x_{t-1} depends on $v_{<t}$ and therefore S_t will depend on it as well. We want S_t to satisfy the following properties (formally stated in Claims 4.3.9–4.3.11):

1. $\mathbb{P}\left\{|\widehat{d_t}| > T \mid A_t \in S_t, v_{<t}\right\} \geqslant 1/6$. In other words, conditioned on S_t, with probability at least 1/6 round t is a update round.
2. Conditioned on S_t *not* occurring, the distribution of v_t under x is *identical* to the distribution of v_t under x', and the privacy loss is 0.
3. Conditioned on S_t, the t-th entry v_t is ε_0 differentially private.

We will define S_t so it contains all of the noise values A_t where $|\widehat{d_t}| = |d_t + A_t|$ is "close to" (within distance σ) or larger than T. This will achieve all three of the above properties. Formally, we construct $S_t = S^+ \cup S^-$ to be made up of two intervals of noise values

$$S^- = (-\infty, -T - d_t + \sigma] \quad \text{and} \quad S^+ = [T - d_t - \sigma, \infty).$$

Note that, since $T > 2\sigma$, these two intervals never intersect. The following claims show that all three properties hold:

Claim 4.3.9 (Property 1). $\mathbb{P}\left\{|\widehat{d_t}| \geqslant T \mid A_t \in S_t, v_{<t}\right\} \geqslant 1/6$.

Proof. Recall that $S^+ = [T - d_t - \sigma, \infty]$. Since A_t is a Laplace random variable with magnitude σ, we get that

$$\mathbb{P}\{A_t \geqslant T - d_t \mid A_t \in S^+, v_{<t}\} = \mathbb{P}\{Lap(\sigma) \geqslant \sigma\} = 1/2e \geqslant 1/6.$$

And similarly, $\mathbb{P}\{A_t \leqslant -T - d_t \mid A_t \in S^-, v_{<t}\} \geqslant 1/6$. Since $|\widehat{d_t}| \geqslant T$ iff $A_t \geqslant T - d_t$ or $A_T \leqslant -T - d_t$, we conclude that $\mathbb{P}\left\{|\widehat{d_t}| \geqslant T \mid A_t \in S_t, v_{<t}\right\} \geqslant 1/6$. ∎

Claim 4.3.10 (Property 2). *For every $a \in \mathbb{R} \cup \{\perp\}$:*

$$\log\left(\frac{P_t(a \mid A_t \notin S_t, v_{<t})}{Q_t(a \mid A_t \notin S_t, v_{<t})}\right) = 0.$$

Proof. When $A_t \notin S_t$, we know that $-T - d_t + \sigma \leqslant A_t \leqslant T - d_t - \sigma$. In particular, this means that (conditioned on S_t *not* occurring), v_t is always $\perp$, both on x and on x'. ∎

Claim 4.3.11 (Property 3). *For every $a \in \mathbb{R} \cup \{\perp\}$:*

$$\log\left(\frac{P_t(a \mid A_t \in S_t, v_{<t})}{Q_t(a \mid A_t \in S_t, v_{<t})}\right) \leqslant 2\varepsilon_0.$$

Proof. Since A_t is a Laplace variable of scale σ, for any $a \in \mathbb{R}$ either its probability by both P_t and Q_t is 0, or otherwise its probabilities by x and x' differ by an $e^{1/\sigma n} = e^{\varepsilon_0}$ ratio. Similarly, we can bound the ratio between the probability of $\perp$ by P and by Q. Note that

$$P_t(\perp \mid A_t, v_{<t}) = \mathbb{P}\{A_t + d_t \in (-T, -T + \sigma] \cup [T - \sigma, T)\},$$

while

$$Q_t(\perp \mid A_t, v_{<t}) = \mathbb{P}\{A_t + d'_t \in (-T, -T + \sigma] \cup [T - \sigma, T)\},$$

where $|d_t - d'_t| \leqslant 1/n$. Since A_t is a Laplacian variable of scale σ, it follows that the ratio of the two probabilities on the RHS is bounded by $e^{1/\sigma n} = e^{\varepsilon_0}$. ∎

Bounding the Expectation. We will now bound the expected loss $\mathbb{E}[L(\mathbf{v})]$ for a random choice of $\mathbf{v}$ sampled according to P. Applying Lemma 2.10.3 to Claim 4.3.11, we get that

$$\mathbb{E}\left[\log\left(\frac{P_t(\mathbf{v}_t \mid A_t \in S_t, \mathbf{v}_{<t})}{Q_t(\mathbf{v}_t \mid A_t \in S_t, \mathbf{v}_{<t})}\right)\right] \leqslant 8\varepsilon_0^2. \tag{4.10}$$

On the other hand, we have by Claim 4.3.10,

$$\mathbb{E}\left[\log\left(\frac{P_t(\mathbf{v}_t \mid A_t \notin S_t, \mathbf{v}_{<t})}{Q_t(\mathbf{v}_t \mid A_t \notin S_t, \mathbf{v}_{<t})}\right)\right] = 0. \tag{4.11}$$

We can express $P_t(\mathbf{v}_t \mid \mathbf{v}_{<t})$ as a convex combination in the form

$$P_t(\mathbf{v}_t \mid \mathbf{v}_{<t}) = \mathbb{P}\{A_t \in S_t \mid \mathbf{v}_{<t}\} P_t(\mathbf{v}_t \mid A_t \in S_t, \mathbf{v}_{<t}\} + \mathbb{P}\{A_t \notin S_t \mid \mathbf{v}_{<t}\} P_t(\mathbf{v}_t \mid A_t \notin S_t, \mathbf{v}_{<t}),$$

and we can express $Q_t(\mathbf{v}_t \mid \mathbf{v}_{<t})$ similarly. We can then apply Lemma 2.10.2 (convexity of relative entropy) to conclude that

$$\mathbb{E}\left[\log\left(\frac{P_t(\mathbf{v}_t \mid v_{<t})}{Q_t(\mathbf{v}_t \mid v_{<t})}\right)\right] \leqslant 8\varepsilon_0^2 \mathbb{E}\left[\mathbb{P}\{A_t \in S_t \mid \mathbf{v}_{<t}\}\right]. \tag{4.12}$$

We conclude that

$$\begin{aligned}
\mathbb{E}\, L(v) &= \sum_{t=1}^{k} \mathbb{E}\left[\log\left(\frac{P_t(\mathbf{v}_t \mid \mathbf{v}_{<t})}{Q_t(\mathbf{v}_t \mid \mathbf{v}_{<t})}\right)\right] \\
&\leqslant 8\varepsilon_0^2 \mathbb{E}\left[\sum_{t=1}^{k} \mathbb{P}\{A_t \in S_t \mid \mathbf{v}_{<t}\}\right] \\
&\leqslant 48\varepsilon_0^2 m \leqslant \varepsilon/2.
\end{aligned} \tag{4.13}$$

Here we used that $\mathbb{E}[\sum_t \mathbb{P}\{A_t \in S_t \mid \mathbf{v}_{<t}\}]$ is just the expected number of borderline rounds which has to be bounded by $6m$ since every borderline round is an update round with probability at least $1/6$ and there are at most m update rounds.

Number of Borderline Rounds. With overwhelming probability, the number m' of borderline rounds (rounds t where S_t occurs) is not much larger than m (the bound on the number of update rounds). This is because every borderline round is with probability at least $1/6$ a update round (Claim 4.3.9). This is made formal in the claim below.

Claim 4.3.12. $\mathbb{P}\left\{m' > 32m\log^{1/2}(1/\delta)\right\} \leqslant \delta/2$

Proof. We have already argued that $\mathbb{E}[m'] \leqslant 6m$. Moreover, the noise in each round is independent from previous rounds. Hence, by tail bounds for Bernoulli variables, the event $m' > 32\sqrt{\log(1/\delta)}m$ has probability less than $\exp(-\log(2/\delta))$. ∎

Putting it together. Condition on there being at most $m' = 32m\log^{1/2}(1/\delta)$ borderline rounds (this is the case with all but $\delta/2$ probability). We proceed by an "evolution of confidence argument" similar to [DN1, DN2].

Specifically, we will apply Azuma's inequality to the set of m' borderline rounds. Formally, let $B \subseteq [k]$ denote the set of borderline rounds. For each $t \in B$, we view

$$X_t = \log\left(\frac{P_t(\mathbf{v}_t \mid \mathbf{v}_{<t})}{Q_t(\mathbf{v}_t \mid \mathbf{v}_{<t})}\right)$$

as a random variable. Note that $L(\mathbf{v}) = \sum_{t\in B} X_t$. Further $|X_t| \leqslant 2\varepsilon_0$ by Claim 4.3.11. Hence, by Azuma's inequality (Lemma 2.10.4),

$$\mathbb{P}\{|L(\mathbf{v})| > \varepsilon\} \leqslant 2\,\mathbb{P}\left\{L(\mathbf{v}) > \mathbb{E}\,L(\mathbf{v}) + \frac{\varepsilon}{2}\right\} \leqslant 2\exp\left(-\frac{\varepsilon^2}{8m'\cdot\varepsilon_0^2}\right).$$

On the other hand, by (4.3),

$$\frac{\varepsilon^2}{8m'\cdot\varepsilon_0^2} \geqslant \frac{\varepsilon^2\sigma^2 n^2}{m'} \geqslant 100\log(1/\delta)\frac{m}{m'}.$$

So, conditioning on having at most m' borderline rounds (occurs with all but $\delta/2$ probability), with all but $\delta/2$ probability the loss $L(\mathbf{v})$ deviates by at most $\varepsilon/2$ from its expectation. The expectation itself is at most $\varepsilon/2$ by (4.13). We conclude that with all but δ probability, the total loss $L(\mathbf{v})$ is bounded by ε in magnitude. ■

4.4 Achieving $(\varepsilon, 0)$-differential privacy

Our previous mechanism satisfies (ε,δ)-differential privacy. We can achieve $(\varepsilon, 0)$-differential privacy (or ε-differential privacy in short) by going from error $n^{-1/2}$ to error $n^{-1/3}$ (in terms of n).

Modifications to PMW. We will need to modify our algorithm in two regards. Specifically, instead of the parameter setting in (4.2) we use the setting

$$\eta = \frac{\log^{1/3} M\cdot\log^{1/3}(k/2\beta)}{\varepsilon^{1/3}n^{1/3}} \qquad \sigma = \frac{10\eta}{\log(k/2\beta)} \qquad T = 40\eta. \tag{4.14}$$

Furthermore, in step (2) of PMW we replace the threshold T by a randomized threshold $\widehat{T} = T + Lap(\sigma_T)$ where $\sigma_T = 10/n\varepsilon$. Our algorithm remains unchanged otherwise.

With these two modifications we can prove the next theorem.

Theorem 4.4.1. *Let $\mathscr{U}$ be a data universe of size N. For any $k, \varepsilon, \beta > 0$, there is an $(\varepsilon, 0)$-differentially private interactive mechanism which is (α, β, k)-accurate for (adaptive) statistical queries over $\mathscr{U}$ and data bases of size n, where*

$$\alpha = O\left(\frac{\log(k/\beta)^{1/3} \log^{1/3} N}{(\varepsilon n)^{1/3}}\right).$$

The running time in answering each query is $N \cdot \mathrm{poly}(n) \cdot \mathrm{polylog}(1/\beta, 1/\varepsilon, 1/\delta)$.

Proof. The algorithm stated in the theorem is given by PMW with the modifications described above. Letting $m = \eta^{-2} \log N$, we note that this setting of parameters in (4.14) satisfies the two properties

$$\sigma n \geqslant \frac{10m}{\varepsilon} \qquad \text{and} \qquad T \geqslant 4\sigma \log(k/\beta).$$

Using the second property, we can repeat the utility analysis verbatim to to argue that there at most $\eta^{-2} \log M$ update rounds. Hence, with probability $1 - \beta/2$, the algorithm answers all k queries with error $\alpha = O(\widehat{T})$. Moreover it is easy to see that with probability $1 - \beta/2$, $\widehat{T} = O(T)$. Hence, with probability $1 - \beta$, we have

$$\alpha = O\left(\frac{\log(k/\beta)^{1/3} \log^{1/3} N}{(\varepsilon n)^{1/3}}\right).$$

It remains to argue that the mechanism satisfies $(\varepsilon, 0)$-differential privacy. Fix two histograms x, x' such that $\|x - x'\|_1 \leqslant 1/n$. As in the proof of Lemma 7.3.4, we let $v \in (\mathbb{R} \cup \{\perp\})^k$ denote a transcript and we let $P(v)$ and $Q(v)$ denote the probability of this transcript when our mechanism is run on x and x', respectively. It suffices to argue that for all transcripts v,

$$-\varepsilon \leqslant \log\left(\frac{P(v)}{Q(v)}\right) \leqslant \varepsilon. \tag{4.15}$$

Let us again write

$$\log\left(\frac{P(v)}{Q(v)}\right) = \sum_{t \in [k]} \log\left(\frac{P_t(v_t \mid v_{<t})}{Q_t(v_t \mid v_{<t})}\right).$$

Further let $H = \{t \in [k] \colon v_t \neq \perp\}$ and $H^c = \{t \in [k] \colon v_t = \perp\} = [k] \backslash H$.

Claim 4.4.2.

$$-\frac{\varepsilon}{10} \leqslant \sum_{t \in H} \log\left(\frac{P_t(v_t \mid v_{<t})}{Q_t(v_t \mid v_{<t})}\right) \leqslant \frac{\varepsilon}{10}. \tag{4.16}$$

Proof. Note that $|H| \leqslant m$ by the termination criterion of our algorithm. It therefore follows from standard properties of the Laplacian distribution and our choice of parameters that

$$\sum_{t \in H} \log\left(\frac{P_t(v_t \mid v_{<t})}{Q_t(v_t \mid v_{<t})}\right) \leqslant \frac{m}{\sigma n} \leqslant \frac{\varepsilon}{10}. \tag{4.17}$$

The same argument shows a lower bound of $-\varepsilon/10$. This concludes the proof. ∎

The next lemma handles the coordinates $t \in H^c$.

Claim 4.4.3.

$$-\frac{\varepsilon}{10} \leqslant \sum_{t \in H^c} \log\left(\frac{P_t(v_t \mid v_{<t})}{Q_t(v_t \mid v_{<t})}\right) \leqslant \frac{\varepsilon}{10}. \tag{4.18}$$

Proof. Let $\mathcal{A}(x)$ be the set of values for the noise variables $(A_1, \ldots, A_k)$ which lead to the event that the transcript is $\perp$ in all rounds $t \in H^c$ when we run our algorithm on input x and condition the transcript on being equal to v_t in all rounds $t \in H$. Define $\mathcal{A}_Z(x)$ in the same way except that we additionally condition the algorithm on the even that $\widehat{T} = Z$. Observe that

$$\mathcal{A}_{Z-1/n}(x') \subseteq \mathcal{A}_Z(x) \subseteq \mathcal{A}_{Z+1/n}(x'). \tag{4.19}$$

Here we used the assumption $\|x - x'\|_1 \leqslant 1/n$ and thus $|f(x) - f(x')| \leqslant 1/n$ for any possible query f.

Further observe that, by the product rule for conditional probabilities,

$$\begin{aligned}\prod_{t \in H^c} P_t(v_t \mid v_{<t}) = \prod_{t \in H^c} P_t(\perp \mid v_{<t}) &= \mathbb{P}\{(A_1, \ldots, A_k) \in \mathcal{A}(x)\} \\ &= \int_{-\infty}^{\infty} \mathbb{P}\left\{\widehat{T} = Z\right\} \mathbb{P}\{(A_1, \ldots, A_k) \in \mathcal{A}_Z(x)\} \, \mathrm{d}Z.\end{aligned}$$

The first step follows from the definition of $\mathcal{A}_Z(x)$ and the second step uses independence between the random variables $\widehat{T}$ and $(A_1, \ldots, A_k)$. On the other hand,

$$\mathbb{P}\left\{\widehat{T} = Z\right\} \leqslant e^{\varepsilon/10} \, \mathbb{P}\left\{\widehat{T} = Z + 1/n\right\},$$

and

$$\mathbb{P}\left\{\widehat{T} = Z - 1/n\right\} \geqslant e^{-\varepsilon/10} \, \mathbb{P}\left\{\widehat{T} = Z - 1/n\right\}.$$

Therefore,

$$\begin{aligned}\prod_{t\in H^c} P_t(v_t \mid v_{<t}) &= \int_{-\infty}^{\infty} \mathbb{P}\left\{\widehat{T} = Z\right\} \mathbb{P}\left\{(A_1,\ldots,A_k) \in \mathcal{A}_Z(x)\right\} \mathrm{d}Z \\ &\leqslant e^{\varepsilon/10} \int_{-\infty}^{\infty} \mathbb{P}\left\{\widehat{T} = Z + 1/n\right\} \mathbb{P}\left\{(A_1,\ldots,A_k) \in \mathcal{A}_{Z+1/n}(x')\right\} \mathrm{d}Z \\ &\qquad\qquad \text{(using (4.19))} \\ &= e^{\varepsilon/10} \int_{-\infty}^{\infty} \mathbb{P}\left\{\widehat{T} = Z\right\} \mathbb{P}\left\{(A_1,\ldots,A_k) \in \mathcal{A}_Z(x')\right\} \mathrm{d}Z \\ &= e^{\varepsilon/10} \prod_{t\in H^c} Q_t(v_t \mid v_{<t}). \end{aligned} \tag{4.20}$$

Using the same reasoning we get

$$\prod_{t\in H^c} P_t(v_t \mid v_{<t}) \geqslant e^{-\varepsilon/10} \prod_{t\in H^c} Q_t(v_t \mid v_{<t}). \tag{4.21}$$

Taking logarithms on both sides of (4.20) and (4.21) shows that (4.18) holds which is what we wanted to show. ■

Putting together (4.16) and (4.18), it follows that the sum over all $t \in [k]$ is in the interval $[-\varepsilon/5, \varepsilon/5]$. This establishes the bound stated in (4.15). We conclude that the algorithm satisfies $(\varepsilon, 0)$-differential privacy. The theorem follows. ■

4.5 Lower bound for $(\varepsilon, 0)$-differential privacy

As discussed before, there is a lower bound of roughly $\sqrt{\log(k)/n}$ that holds for blatant non-privacy [DN1]. A shortcoming is that this lower bound does not depend on the universe size. In this section we will show a lower bound on the accuracy of any mechanism that satisfies $(\varepsilon, 0)$-differential privacy even if it works in the non-interactive setting.

Theorem 4.5.1. *Let n be sufficiently large and let $\varepsilon > 0$ be a constant independent of n. Then, for every $k \geqslant n^{1.1}$ there is a set of k linear queries over a universe of size N such that every $(\varepsilon, 0)$-differentially private mechanism for databases of size n must have error*

$$\alpha \geqslant \Omega(1) \cdot \left(\frac{\log k \cdot \log\left(\frac{N}{n}\right)}{\varepsilon n} \right)^{1/2}$$

with probability 1/2.

Remark 4.5.2. We remark that the requirement $k \geqslant n^{1.1}$ can be replaced by $k \geqslant n^{1+c}$ for every constant $c > 0$.

Proof. Let $\mathscr{U}$ be a universe of size N. Our lower bound uses a discrete variant of the packing argument from [HT].

Consider the family $\mathcal{X} \subseteq (\frac{1}{n}\mathbb{Z}_+)^N$ of all histograms with exactly s nonzeros. Note that such histograms correspond to databases of size n with s distinct elements. Here, s is some parameter that we will fix shortly. Since we normalize histograms to have norm 1, this means that each nonzero coordinate is $1/s$. Further let $\mathcal{F}$ be the uniform distribution over linear queries of the form $f \in \{0,1\}^N$.

We say that two histograms $x, y \in \mathcal{X}$ are *half disjoint* if $\|x - y\|_1 \geqslant 1/2$. The next claim shows that two randomly chosen elements from $\mathcal{X}$ are very likely half disjoint.

Claim 4.5.3. *The probability that $x, y \sim \mathcal{X}$ are not half disjoint is at most* $\exp(-\Omega(s \cdot \log(N/n)))$.

Proof. Note that for $x, y \sim \mathcal{X}$ not to be half disjoint it must be the case that half the nonzero coordinates of y must fall into the s nonzero coordinates of x. The probability of this event is less than

$$\binom{s}{s/2} \cdot \left(\frac{s}{N}\right)^{s/2} \leqslant 2^{-\Omega(s \log(N/s))}.$$

Here we used that $s \leqslant n$. ∎

We also need to show that any two half disjoint histograms x, y are "well separated" by k random linear queries.

Claim 4.5.4. *Let x, y be half disjoint. Then, for k queries $f_1, \ldots, f_k$ chosen uniformly at random from $\mathcal{F}$, we have*

$$\mathbb{P}\left\{\max_{t \in [k]} \left|f_t(x) - f_t(y)\right| \leqslant \frac{c}{\sqrt{s}}\right\} \leqslant \exp\left(-2^{-\Omega(c^2)}k\right). \tag{4.22}$$

Proof. If we choose a single query f at random, then $|f(x) - f(y)|$ is expected to be $\Theta(1/\sqrt{s})$. Further, $|f(x) - f(y)| > c/\sqrt{s}$ with probability at least $2^{-\Omega(c^2)}$. This follows from standard lower bounds on the tail of the binomial distribution. The probability that none out of k random queries has difference $c/\sqrt{s}$ is therefore bounded by $(1 - 2^{-\Omega(c^2)})^k$ which implies the claim. ∎

We will now put the previous two claims together. To this end, fix $s = C\varepsilon n/\log(N)$ for sufficiently large $C > 0$ and put

$$\alpha_0 = \sqrt{\frac{\log k}{s}} = \sqrt{\frac{\log k \cdot \log N}{\varepsilon n}}. \tag{4.23}$$

It then follows from Claim 4.5.4 that

$$\mathbb{P}\left\{\max_{t\in[k]}\left|f_t(x)-f_t(y)\right|\leqslant\alpha_0\right\}\leqslant\exp\left(-k^{0.99}\right). \tag{4.24}$$

On the other hand, using Claim 4.5.3, it follows that there exists a set $\mathcal{P}\subseteq\mathcal{X}$ such that every pair $x,y\in\mathcal{P}$ with $x\neq y$ is half disjoint and

$$|\mathcal{P}|\geqslant\exp\left(\Omega\left(s\log(N/n)\right)\right)\geqslant 3\exp(\varepsilon n). \tag{4.25}$$

In the second inequality we used the fact that we can choose C sufficiently large in the setting of s above.

Further, by our assumption $k\geqslant n^{1.1}$, we have $\exp(-k^{0.99})\ll|\mathcal{P}|^{-2}$. Hence, we can take the union bound over all distinct pairs in $\mathcal{P}$ and conclude that there must exist a set of k linear queries $f_1,\ldots,f_k$ such that for every two histograms $x,y\in\mathcal{P}$ with $x\neq y$ we have

$$\max_{t\in[k]}\left|f_t(x)-f_t(y)\right|\geqslant\alpha_0.$$

Now, for the sake of contradiction suppose there is an $(\varepsilon,0)$-differentially private mechanism M for answering the k linear queries $f_1,\ldots,f_k$ which has maximum error $\alpha=\alpha_0/2$ with probability 1/2.

For a histogram x let Fx denote the vector $(f_1(x),\ldots,f_k(x))\in\mathbb{R}^k$ and let $B(Fx,\alpha)$ denote the ℓ_∞-ball around Fx of radius α. Note that by the accuracy gurantee of M, we have for every histogram x,

$$\mathbb{P}\left\{M(x)\in B(Fx,\alpha)\right\}\geqslant\frac{1}{2}.$$

By $(\varepsilon,0)$-differential privacy, we further have for every two histograms x,y,

$$\mathbb{P}\left\{M(y)\in B(Fx,\alpha)\right\}\geqslant\frac{\exp(-\varepsilon n)}{2}.$$

Fix any histogram $x\in\mathcal{P}$, then

$$\begin{aligned}1\geqslant\mathbb{P}\left\{M(x)\in\bigcup_{y\in\mathcal{P}}B(Fy,\alpha)\right\}&=\sum_{y\in\mathcal{P}}\mathbb{P}\left\{M(x)\in B(Fy,\alpha)\right\}\\&\geqslant|\mathcal{P}|\frac{\exp(-\varepsilon n)}{2}.\end{aligned}$$

Here we used that for $y,y'\in\mathcal{P}$ with $y\neq y'$ we have $B(Fy,\alpha)\cap B(Fy',\alpha)=\emptyset$. But $|\mathcal{P}|\geqslant 3\exp(\varepsilon n)$. Hence we have arrived at a contradiction showing that such a mechanism M cannot exist. ∎

Remark 4.5.5. In our proof we used databases in which individual data items occur with high multiplicity. This is not necessary as we can always move to a universe of size nN in which every data item occurs n times. Hence, we can repeat the same construction without multiplicities by losing only a factor of n in the universe size.

We leave it as an open problem to close the gap between our upper and lower bound. Indeed, it would be interesting to know if the optimal dependence on n is $n^{-1/3}$ or rather $n^{-1/2}$. We also point out the open problem of coming up with a lower bound for (ε, δ)-differential privacy, such as $\Omega(\log^{c_1} k \cdot \log^{c_2} N \cdot n^{-1/2})$, that simultaneously has a (poly-logarithmic) dependence on the universe size N and the number of queries k.

4.6 Average-case complexity and smooth instances

In this section, we define a notion of average case complexity for interactive (and non-interactive) mechanisms that allows us to improve the running time of the PMW mechanism as a function of the data universe size. This is done using an argument for reducing the data universe size.

We start by defining the notion of a *smooth* histogram. We think of these histograms as distributions over the data universe that do not place too much weight on any given data item. In other words, we require the histogram to have high min-entropy.

Definition 4.6.1 (Smooth)**.** A histogram $x \in \mathbb{R}^{\mathscr{U}}$ s.t. $\sum_{u \in \mathscr{U}} x_u = 1$ and $\forall u \in \mathscr{U} : x_u \geqslant 0$ is ξ-smooth if $\forall u \in \mathscr{U} : x_u \leqslant \xi$.

In particular, a ξ-smooth histogram has *min-entropy* at least $\log(1/\xi)$. We typically think of ξ as a function of N, such as $polylogN/N$ or $1/\sqrt{N}$. Note that small databases (viewed as histograms) cannot be very smooth, since a ξ-smooth histogram has at least $1/\xi$ nonzero coordinates.

We therefore extend the notion of smoothness to the notion of pseudo-smoothness with respect to a set of queries $\mathcal{Q}$. A histogram is *pseudo-smooth* w.r.t a query class $\mathcal{Q}$ roughly speaking when there exists a smooth histogram x^* that is close on every query in $\mathcal{Q}$. This notion allows even very sparse histograms (corresponding to small databases) to be very pseudo-smooth. The formal definition is as follows.

Definition 4.6.2 (Pseudo-smooth)**.** A histogram $x \in \mathbb{R}^{\mathscr{U}}$ s.t. $\sum_{u \in \mathscr{U}} x_u = 1$ and $\forall u \in \mathscr{U} : x_u \geqslant 0$ is (ξ, ϕ)-smooth w.r.t a class of linear queries $\mathcal{Q}$ if there *exists* a ξ-smooth histogram x^* s.t.

$$\forall f \in \mathcal{Q}: \qquad |f(x) - f(x^*)| \leqslant \phi.$$

A straightforward way of obtaining pseudo-smooth databases is by sampling from a smooth histogram.

Claim 4.6.3. *Let $\mathcal{U}$ be a data universe, $\mathcal{Q}$ a class of linear queries over $\mathcal{U}$, and x^* a ξ-smooth histogram over $\mathcal{U}$. For any $\alpha, \beta > 0$, sample a database x of $m = (\log(2/\beta) + \log|\mathcal{Q}|)/\alpha^2$ items i.i.d from the distribution of x^* (i.e. in each sample we independently pick each $u \in \mathcal{U}$ with probability x^*_u). Then with all but β probability over the samples taken, $\forall f \in \mathcal{Q}: |f(x) - f(x^*)| \leqslant \alpha$, and so the database x is (ξ, α)-pseudo-smooth w.r.t $\mathcal{Q}$.*

Proof. The proof is by a Chernoff bound (as in [DNR+]). ∎

4.6.1 Domain reduction for pseudo-smooth histograms

For a given smoothness parameter ξ, data universe $\mathcal{U}$, and query class $\mathcal{Q}$, let $V \subseteq \mathcal{U}$ be a sub-universe sampled uniformly and at random from $\mathcal{U}$. In this section we show that (as long as V is large enough) if x was a pseudo-smooth histogram over $\mathcal{U}$ w.r.t a query class $\mathcal{Q}$, then w.h.p. there will be a histogram x^* *with support only over (the smaller)* V that is "close" to x on $\mathcal{Q}$. We emphasize that sampling the sub-universe V does not require knowing x nor knowing any x^* that certifies x being pseudo-smooth, we only need to know ξ. In particular, this approach is privacy-preserving. This technique for reducing the universe size can be used to improve the efficiency of the PMW mechanism for pseudo-smooth input databases.

Lemma 4.6.4. *Let $\mathcal{U}$ be a data universe and $\mathcal{Q}$ a collection of linear queries over $\mathcal{U}$. Let x be (ξ, ϕ)-pseudo-smooth w.r.t $\mathcal{Q}$. Take $\alpha, \beta > 0$, and sample uniformly at random (with replacement) $V \subseteq \mathcal{U}$ so that*

$$M = |V| = 4\max\{\xi N \cdot (\log(1/\beta) + \log|\mathcal{Q}|)/\alpha^2, \log(1/\beta)\} \tag{4.26}$$

Then, with all but β probability over the choice of V, there exists a histogram x^* with support only over V such that

$$\forall f \in \mathcal{Q}: |f(x) - f(x^*)| \leqslant \phi + \alpha. \tag{4.27}$$

Proof. Let y be the ξ-smooth histogram which shows that x is (ξ, ϕ)-pseudo-smooth. If we sampled uniformly at random from x or from y then by Claim 4.6.3, we could get a database over a very small sub-universe that is (as required) close to x on all the queries in $\mathcal{Q}$. This is insufficient because we want the sub-universe that we find to be *independent* of the database x (and so also independent of y).

Still, let us re-examine the idea of sampling from y. One way of doing this is by *rejection sampling*. Namely, repeatedly sample $u \in \mathcal{U}$ *uniformly at random*

and then "keep" u with probability y_u/ξ. Otherwise reject. When we use this rejection sampling, since y is a ξ-smooth distribution, each sample that we keep is distributed by y (i.e. it is $u \in \mathcal{U}$ w.p. y_u). Repeat this process until $m_1 = (\log(2/\beta) + \log|\mathcal{Q}|)/\alpha^2$ samples have been accepted. There is now a set of coordinates $V_1 \subseteq \mathcal{U}$, those that were kept (of size at most m_1), and a set of coordinates $V_2 \subseteq \mathcal{U}$, those that were rejected. By Claim 4.6.3 the sub-universe V_1 of samples that we keep (which are i.i.d samples from y) supports (except with probability $\beta/2$) a database x^* that is "close" to y (w.r.t $\mathcal{Q}$), and so it will also be "close" to x. In particular, by triangle inequality,

$$\max_{f\in\mathcal{Q}} |f(x) - f(x^*)| \leqslant \max_{f\in\mathcal{Q}} |f(x) - f(y)| + \max_{f\in\mathcal{Q}} |f(y) - f(x^*)| \leqslant \phi + \alpha\,.$$

But now we may take $V = V_1 \cup V_2$. Note that V is simply a uniformly random subset of the coordinates of $\mathcal{U}$. And by the previous argument, V supports a histogram that satisfies (4.27), namely x^*. To conclude the proof it remains to argue that V has the required size. Note that the probability of accepting sample i in the rejection procedure is given by $\sum_{i=1}^{N} \frac{1}{N} \cdot \frac{y_i}{\xi} = 1/\xi N$. Hence, the expected number of queries in total is $\mu = 2\xi N \cdot (\log(2/\beta) + \log|\mathcal{Q}|)/\alpha^2$. Moreover, since every sample is independent, we have concentration around the expectation. A multiplicative Chernoff bound shows that the probability that V is larger than twice its expectation is bounded by $\exp(-\mu) \leqslant \beta/2$.

■

Finally, we use Lemma 4.6.4 together with Lemma 4.3.7 (utility of PMW for general V), to derive the accuracy guarantee of Theorem 4.1.4 for the performance of the PMW mechanism on pseudo-smooth databases.

Proof of Utility for Theorem 4.1.4. We run PMW on a uniformly chosen sub-universe V of the appropriate size M as stated in Equation (4.26) above, taking $\alpha = 1/\sqrt{n}$. We conclude that with all but $\beta/2$ probability over the sampling, there exists a database x^* supported on V that is $\phi + 1/\sqrt{n}$-close to x w.r.t. the given sequence of k statistical queries. Plugging this into Lemma 4.3.7, we obtain the accuracy bound claimed in Theorem 4.1.4. ■

Chapter 5

A Simple and Practical Non-Interactive Release Mechanism

Applications of the theory of differentially private data analysis to real-world data have received less attention in the past, and have so far met with mixed success. Eager data analysts have on some applications found that the previous state of the art in algorithms for achieving differential privacy add unacceptable levels of noise; such situations are not evidence that differential privacy is an inappropriate definition, but rather that we need to develop and implement better algorithms, perhaps tailored to the application at hand. In many cases, existing theoretical results have been focused on demonstrating good asymptotic worst-case behavior, but with little regard for constant factors or performance on realistic database sizes.

In this chapter we consider the multiplicative weights approach in the *non-interactive* query release setting. While our algorithm from the previous chapter could be used directly in the non-interactive setting, we will take a different route. Specifically, we will give a simplified algorithm with a much simpler privacy analysis that achieves the same bounds in the non-interactive setting as we did earlier in the interactive setting. Moreover, we successfully evaluate our algorithm on a variety of real world data sets.

Our approach uses the exponential mechanism [MT] as a subroutine to multiplicative weights framework from the previous chapter. On a high-level, the multiplicative weights approach maintains a candidate output at all times, represented as a distribution over the space of possible data points. The quality of the distribution when compared with the true database is repeatedly improved by a procedure that selects a query from the target class, and reweights the distribution to improve its fidelity on the selected query. In our algorithm, presented in Section 5.2, we use the exponential mechanism as a means to bias our selection of the next query towards one that will provide the most improvement to our distribution.

Experimentally, we test our algorithm on four standard data sets, the first three of which are also studied by Fienberg et al. [FRY] (see their paper for further discussion of the data sets and references to additional work on this data):

1. an epidemiological study of Czechoslovakian car factory workers intended to investigate risk factors for coronary thrombosis;

2. a 1990 genetic study of barley powdery mildew isolates using DNA markers;

3. data relating household characteristics, women's economic activity, and husband's unemployment, in households in the city of Rochdale; and

4. the National Long Term Care Survey, a longitudinal study of the health of older Americans, based a sample of tens of thousands of Medicare enrollees.[1]

On each of these data sets, we present experimental results that demonstrate the tradeoff between the differential privacy parameter and the accuracy (as measured by relative entropy) of the resulting data, comparing the results when privacy is achieved by each of three different differentially private algorithms: (1) the original approach of Barak et al. [BCD+] for producing synthetic contingency table data (this approach is analogous to the experiments undertaken in Fienberg et al. [FRY]), (2) our algorithm combining multiplicative weights with the exponential mechanism as described above, with no specialized optimization, and (3) our algorithm, with a number of additional optimizations described in Section 5.3.

5.1 Main results

We will state our main theorems informally here. A formal statement is given in Section 5.2.4. As in the previous chapter we consider databases of size n over a universe $\mathcal{U}$ of size N represented by a histogram $x \in \mathbb{R}^N$ normalized such that $\sum_{u \in \mathcal{U}} x_u = 1$. With this normalization we will think of x as a distribution over the universe. The queries asked by the analyst are linear queries $f \in [0,1]^N$ so that $f(x) = \langle f, x \rangle$. A feature of our algorithms is that they produce *synthetic data*. That is the output is a histogram $x^* \in \mathbb{R}^N$. The accuracy requirement is that of Definition 2.3.1, namely, we require that the expectation of $\max_{f \in \mathcal{Q}} |f(x) - f(x^*)|$ is bounded by α where $\mathcal{Q}$ is a given set of linear queries.

Theorem 5.1.1 (informal). *Given a data set x of size n over a universe $\mathcal{U}$ of size N and given a set of statistical queries $\mathcal{Q}$ of size k, we can compute an* $(\varepsilon, 0)$-

[1] See http://www.nltcs.aas.duke.edu/ for more details on the survey and associated data.

differentially private data distribution x^* that is α-accurate with

$$\alpha \leqslant O\left(\frac{\log N \log k}{\varepsilon n}\right)^{1/3}.$$

We can obtain a stronger bound on the error by allowing (ε, δ)-differential privacy.

Theorem 5.1.2 (informal). *Given a data set x of size n over a universe $\mathcal{U}$ of size N and given a set of statistical queries $\mathcal{Q}$ of size k, we can compute an (ε, δ)-differentially private data distribution x^* that is α-accurate with*

$$\alpha \leqslant O\left(\frac{\sqrt{\log N \log(1/\delta)} \log k}{\varepsilon n}\right)^{1/2}.$$

The probability that the mechanism fails to achieve the stated error bound can be bounded by $1/\mathrm{poly}(k)$ without changing the stated bounds as shown in Section 5.2.1. It can be reduced even further at a small loss in accuracy. The exact dependence is omitted from the theorems for simplicity. We also remark that the runtime of our algorithm nearly matches the cryptographic hardness results of [DNR+] as our algorithm produce synthetic data.

Experimental evaluation. Our experimental observations bear out the significance of choosing to take only the most significant measurements, at improved accuracy. On several real data sets, our algorithms yield marked improvement over the prior naive approaches of taking all measurements one seeks to preserve. The improvement is most significant when privacy constraints are strong and the query class is rich; in applications where one can afford to simply take all measurements at sufficient accuracy, careful selection is not helpful. Fortunately, the former setting is the most important and challenging for resolving the practical tension between privacy and utility. A detailed discussion is presented in Section 5.3.

5.1.1 Comparison to previous work

The work of Barak et al. [BCD+] was the first to address the problem of generating synthetic databases that preserve differential privacy. Their algorithm, which maintains utility with respect to a set of marginals (as opposed to general statistical queries), essentially computes the desired noisy marginals and then solves the linear program constrained by these noisy marginals in order to obtain consistent data. This approach identifies all measurements required to reproduce the marginals, and takes each with a uniform level of

accuracy. This may make a large number of redundant or uninformative measurements at the expense of accuracy in the more interesting queries. Fienberg et al. [FRY] observe that, on realistic data sets, the Barak et al. algorithm must add so much noise to preserve differential privacy that the resulting data is no longer useful.

The study of differentially private synthetic data release mechanisms for arbitrary statistical queries began with the work of Blum, Ligett, and Roth [BLR], who gave a computationally inefficient (superpolynomial in $N = |\mathscr{U}|$) algorithm that achieves error that scales only logarithmically with the number of queries. The dependence on the size of the data set n achieved by their algorithm is $n^{-1/3}$.

Li et al. [LHR+] investigated an approach to answering sets of statistical queries, which selects an appropriate basis to cover the target set of statistical queries and reconstructs answers from this basis. They are primarily concerned with settings of $k \gg n$. Our algorithm, when applied to the specific query sets for which they state results, reduces the maximum error rates substantially. For example, for the case of $k = 2^N$ (corresponding to the set of all $(0,1)$-statistical queries), the dependence on N goes from $N \log^2 n$ to $N^{1/3} \log^{1/3} N$.

Since [BLR], subsequent work has focused on computationally more efficient algorithms (here meaning polynomial in N). This line of work has yielded error rates of of $(k^{o(1)}/\sqrt{n})$ [DNR+] and $\mathrm{poly}(\log k)/\sqrt{n}$ [DRV] for a relaxed privacy guarantee known as (ε, δ)-differential privacy. Here, k represents the number of queries $|\mathcal{Q}|$. The private multiplicative weights framework of Chapter 4 shows error rates of $\log k/\sqrt{n}$ for (ε, δ)-differential privacy. Still, the algorithm can also be used non-interactively to produce synthetic data. Indeed, by asking the entire set of queries roughly n times repeatedly, one can ensure that on at least one of the iterations there are no large errors. In this case the multiplicative weights distribution represents synthetic data that is correct (up to the desired error) on all k queries.

We build on Chapter 4 for the task of creating synthetic data. (Our results can be interpreted as providing synthetic data, by means of sampling the private distribution we generate.) In addition, our algorithm and its analysis are both significantly simpler. Further, unlike all the previous work mentioned above, our new results are tuned for practical applications, and we arguably provide the first empirical results demonstrating that it is possible to produce useful differentially private synthetic data for real-world statistical applications.

Our algorithm can also be seen as an instance of a more general framework due to [GHRU] which we will present in Chapter 6. Specifically, we will see that any agnostic learning algorithms can be used as a subroutine in the

multiplicative weights framework to select queries with near maximal error. Without defining what an agnostic learning algorithm is at this point, we mention that the exponential mechanism can be considered one as discussed in [KLN+].

5.2 Multiplicative weights with exponential mechanism

Our algorithm is presented in Figure 5.1. Its utility guarantees are stated and analyzed in Section 5.2.1. The privacy analysis follows in Section 5.2.3. Finally, we state and prove our privacy-utility trade-off theorems in Section 5.2.4.

Input: Parameters $T \in \mathbb{N}, \varepsilon_0 > 0$, distribution x over a universe $\mathscr{U}$
Let x_0 denote the uniform distribution over $\mathscr{U}$, i.e., $x_0[u] = 1/N$ for all $u \in \mathscr{U}$.
For $t = 1, ..., T$ iterations:

1. *Exponential Mechanism:* Sample a query $f_t \in \mathcal{Q}$ from the exponential mechanism defined on $\mathcal{Q}$ using the score function $n|d_t(f)|/2$ where $d_t(f) = f(x) - f(x_{t-1})$. That is, each f has probability proportional to $\exp(\varepsilon_0 n|d_t(f)|/2)$.
2. Compute $\widehat{d_t}(f_t) = d_t(f_t) + Lap(2/\varepsilon_0 n)$.
3. *Multiplicative Weights Update:* Let x_t be the distribution obtained from x_{t-1} using a multiplicative weights update step as $x'_t[u] = \exp(\widehat{d_t}(f_t))x_{t-1}[u]$, and renormalize $x_t[u] = \frac{x'_t[u])}{\sum_{u \in \mathscr{U}} x'_t[u]}$.

Output: $x^* = \frac{1}{T}\sum_{t=1}^{T} x_t$.

Figure 5.1: Multiplicative weights update with exponential mechanism (Algorithm 5.1)

Comparing Algorithm 5.1 with Algorithm 4.2, we can see that here we do not use a fixed update parameter. Rather the update parameter is a random variable $\widehat{d_t}(f_t)$. This change turned out to yield the best experimental results though it does not lead to an asymptotic improvement compared to an optimally calibrated fixed parameter. This change manifests itself in the utility analysis in that we will have to bound the *expected* potential drop.

5.2.1 Utility Analysis

We denote the worst-case (or maximum) error of our output over all queries by

$$\operatorname{err}(x^*, \mathcal{Q}) \stackrel{\text{def}}{=} \max_{f \in \mathcal{Q}} |f(x^*) - f(x)| . \tag{5.1}$$

The utility guarantee of our algorithm is captured by the following lemma that gives a bound on the expected value of $\operatorname{err}(x^*, \mathcal{Q})$.

Lemma 5.2.1 (Expected Maximum Error).

$$\mathbb{E} \operatorname{err}(x^*, \mathcal{Q}) \leqslant O\left(\sqrt{\frac{\log N}{T}} + \frac{\log k}{\varepsilon_0 n} \right) \tag{5.2}$$

Here the expectation is taken over the randomness of Algorithm 5.1.

We can show a large deviation bound on $\operatorname{err}(x^*, \mathcal{Q})$ as stated in the following lemma.

Lemma 5.2.2 (Concentration). *Let α denote the right hand side in Equation 5.2. Then, there is a constant $C > 0$ such that for every $\ell > 0$,*

$$\mathbb{P}\left\{ \operatorname{err}(x^*, \mathcal{Q}) > \alpha + \frac{\ell}{\varepsilon_0 n} \right\} \leqslant T \exp(-C\ell). \tag{5.3}$$

In typical settings $\log T = O(\log k)$. In this case by setting $\ell = O(\log k)$, the lemma allows us to bound the failure probability by $1/\mathrm{poly}(k)$ without increasing the error by more than a constant factor. The proof of this lemma is given in Section 5.2.2. For simplicity we will not include the failure probability as an explicit parameter in our theorems later.

Proof of Lemma 5.2.1. Let us use the shorthand $\operatorname{err}_t \stackrel{\text{def}}{=} \operatorname{err}(x_t, \mathcal{Q}) = \max_{f \in \mathcal{Q}} |d_t(f)|$ to denote the worst-case error at step t of our algorithm. As we saw in Lemma 2.5.2, the exponential mechanism at step t selects a query whose error nearly matches err_t. For convenience, we restate this lemma here specialized to our setting.

Lemma 5.2.3 ([MT]). *For every $t \in \{1, \ldots, T\}$,*

$$\mathbb{P}\{|d_t(f_t)| \leqslant \operatorname{err}_t - r\} \leqslant \frac{k \exp(-\varepsilon_0 r n/2)}{|\{f \in \mathcal{Q} \colon d_t(f_t) > \operatorname{err}_t - r\}|}.$$

Lemma 5.2.4.

$$\mathbb{E} |d_t(f_t)| \geqslant \operatorname{err}_t - \frac{2 \log k + 1}{\varepsilon_0 n}.$$

Proof. Note that the denominator in the RHS of Lemma 5.2.3 is always at least 1. Hence, by Lemma 5.2.3,

$$\mathbb{P}\left\{|d_t(f_t)| \leqslant \mathrm{err}_t - \frac{2\log k + \ell}{\varepsilon_0 n}\right\} \leqslant \exp(-\ell). \tag{5.4}$$

On the other hand $\int_0^\infty \ell \exp(-\ell)\mathrm{d}\ell = 1$. ∎

We will next analyze the convergence of our algorithm to a good distribution using a similar potential argument as we saw in Chapter 4. The difference here is that the update parameter in our algorithm is randomized and hence we will compute the expected potential drop. Specifically, we show that while our error bounds are not met, each update results in a significant decrease in the relative entropy of x_t and x, which is initially at most $\log N$ and always at least 0. This bounds the number of rounds before the error bounds become satisfied.

We consider the potential function $\Psi_t = \mathrm{RE}(x\|x_t)$. The following two properties follow from non-negativity of entropy, and Jensen's Inequality:

Fact 5.2.5. $\Psi_t \geqslant 0$

Fact 5.2.6. $\Psi_0 \leqslant \log N$

The next lemma gives a lower bound on the expected potential drop.

Lemma 5.2.7. *In expectation over the Laplacian random variable at step t, we have*

$$\mathbb{E}\left[\Psi_{t-1} - \Psi_t\right] \geqslant \frac{1}{2}|d_t(f_t)|^2 - \frac{1}{4}\mathbb{E}\left|\widehat{d_t}(f_t)\right|^2$$

Proof. Lemma 4.3.3 from Chapter 4 shows that

$$\Psi_{t-1} - \Psi_t \geqslant \eta d_t(f_t) - \eta^2$$

where η is a scaling parameter that appears in the multiplicative weights update. We chose $\eta = 1/2\widehat{d_t}(f_t)$ so that

$$\Psi_{t-1} - \Psi_t \geqslant \frac{1}{2}\widehat{d_t}(f_t)\cdot d_t(f_t) - \frac{1}{4}\left|\widehat{d_t}(f_t)\right|^2.$$

Taking expectations over the Laplacian random variable in step t, we get

$$\mathbb{E}\left[\widehat{d_t}(f_t)\cdot d_t(f_t)\right] = d_t(f_t)\,\mathbb{E}\,\widehat{d_t}(f_t) = d_t(f_t)\cdot d_t(f_t).$$ ∎

Lemma 5.2.8. $\mathbb{E}\left|\widehat{d_t}(f_t)\right|^2 \leqslant |d_t(f_t)|^2 + \frac{8}{\varepsilon_0^2 n^2}$

Proof. Recall that $\widehat{d_t}(f_t) = d_t(f_t) + Lap(2/\varepsilon_0 n)$. The claim now follows from the fact that $\mathbb{E}\, Lap(\sigma)^2 = 2\sigma^2$ ∎

We will now compute the expected potential drop where this time the expectation is taken over the entire randomness of our algorithm. This will allow us to sum the total expected potential drop over all steps of our algorithm.

Combining the previous two lemmas, we get

$$\mathbb{E}\,[\Psi_{t-1} - \Psi_t] \geqslant \frac{1}{4}\,\mathbb{E}\,|d_t(f_t)|^2 - \frac{2}{\varepsilon_0^2 n^2}. \tag{5.5}$$

On the other hand, by Jensen's inequality and Lemma 5.2.4,

$$\mathbb{E}\,|d_t(f_t)|^2 \geqslant \left(\mathbb{E}\,|d_t(f_t)|\right)^2 \geqslant \left(\mathbb{E}\,\mathrm{err}_t - \frac{2\log k + 1}{\varepsilon_0 n}\right)^2 \tag{5.6}$$

Combining (5.5) with (5.6), we get

$$\mathbb{E}\,[\Psi_{t-1} - \Psi_t] \geqslant \frac{1}{4}\left(\mathbb{E}\,\mathrm{err}_t - \frac{2\log k + 1}{\varepsilon_0 n}\right)^2 - \frac{2}{\varepsilon_0^2 n^2}. \tag{5.7}$$

By linearity of expectation, Fact 6.5.4, and Fact 6.5.5, we have

$$\sum_{t=1}^{T} \mathbb{E}\,[\Psi_{t-1} - \Psi_t] = \mathbb{E}\left[\sum_{t=1}^{T} \Psi_{t-1} - \Psi_t\right] = \mathbb{E}\,[\Psi_0 - \Psi_T] \leqslant \log N.$$

Therefore,

$$\sum_{t=1}^{T}\left(\mathbb{E}\,\mathrm{err}_t - \frac{2\log k + 1}{\varepsilon_0 n}\right)^2 \leqslant 4\log N + \frac{8}{\varepsilon_0^2 n^2}. \tag{5.8}$$

On the other hand, by Cauchy-Schwarz ($\sum a_t^2 \geqslant \frac{1}{T}(\sum a_t)^2$),

$$\sum_{t=1}^{T}\left(\mathbb{E}\,\mathrm{err}_t - \frac{2\log k + 1}{\varepsilon_0 n}\right)^2 \geqslant \frac{1}{T}\left(\sum_{t=1}^{T}\left[\mathbb{E}\,\mathrm{err}_t - \frac{2\log k + 1}{\varepsilon_0 n}\right]\right)^2. \tag{5.9}$$

Combining (5.8) with (5.9) and rearranging, we get

$$\sum_{t=1}^{T} \mathbb{E}\,\mathrm{err}_t \leqslant \sqrt{4T\log N + \frac{8T}{\varepsilon_0^2 n^2}} + \frac{T(2\log k + 1)}{\varepsilon_0 n}.$$

Lemma 5.2.1 now follows by observing that

$$\mathbb{E}\,\mathrm{err}(x^*, \mathcal{Q}) \leqslant \sum_{t=1}^{T} \frac{\mathbb{E}\,\mathrm{err}_t}{T} \leqslant \sqrt{\frac{4\log N}{T} + \frac{8}{T\varepsilon_0^2 n^2}} + \frac{2\log k + 1}{\varepsilon_0 n}. \tag{5.10}$$

We can further simplify this bound by noting that $T \geqslant 1$ and $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$ for non-negative a, b. Hence, the additional term of $8/T\varepsilon_0^2 n^2$ under the square root is bounded by $O(1/\varepsilon_0 n) \leqslant O(\log k/\varepsilon_0 n)$. This concludes the proof of Lemma 5.2.1.

5.2.2 A large deviation bound

It is not difficult to show that $\mathrm{err}(x^*, \mathcal{Q})$ is unlikely to be significantly larger than its expectation.

Proof of Lemma 5.2.2. Let X_t denote the deviation of the exponential mechanism from err_t in round t of our algorithm. That is $X_t = \mathrm{err}_t - d_t(f_t)$. Note that $X_t \geqslant 0$. As was argued in Equation 5.4, we have that $\mathbb{P}\{X_t > (2\log k + \ell)/\varepsilon_0 n\} \leqslant \exp(-\ell)$. Likewise, let $Y_t \geqslant 0$ denote the deviation of the Laplace mechanism from its mean at step t. That is $Y_t = |d_t(f_t) - \widehat{d_t}(f_t)|$. By basic properties of the Laplace distribution, we have $\mathbb{P}\{Y_t > \ell/\varepsilon_0 n\} \leqslant \exp(-\ell/10)$. Finally, let $Z_t = X_t + Y_t$.

With this notation we can follow the proof of Lemma 5.2.1 step by step, but without taking expectations. We can replace Equation 5.8 by

$$\sum_{t=1}^{T} (\mathrm{err}_t - X_t)^2 \leqslant 4\log N + 4\sum_{t=1}^{T} Y_t^2. \tag{5.11}$$

From this we conclude,

$$\sum_{t=1}^{T} \mathrm{err}_t \leqslant \sqrt{4\log N + 4\sum_{t=1}^{T} Y_t^2} + \sum_{t=1}^{T} X_t \leqslant \sqrt{4\log N} + 2\sum_{t=1}^{T} Y_t + \sum_{t=1}^{T} X_t, \tag{5.12}$$

using the fact that $\sqrt{\sum_t Y_t^2} \leqslant \sum_t Y_t$. Therefore,

$$\mathrm{err}(x^*, \mathcal{Q}) \leqslant O\left(\sqrt{\frac{\log N}{T}} + \frac{1}{T}\sum_{t=1}^{T} Z_t\right). \tag{5.13}$$

Combining this with our previous observations and taking the union bound over all variables Z_t, there is a constant $C' = O(1)$ such that

$$\mathbb{P}\left\{\mathrm{err}(x^*, \mathcal{Q}) > C'\left(\sqrt{\frac{\log N}{T}} + \frac{\log k + \ell}{\varepsilon_0 n}\right)\right\} \leqslant T\exp(-\ell).$$

Comparing this with our bound on $\mathbb{E}\,\mathrm{err}(x^*, \mathcal{Q})$ (as given in Lemma 5.2.1), the claim follows.

5.2.3 Privacy Analysis

Our privacy analysis can be derived easily from the composition theorems stated in Theorem 2.4.4 and Theorem 2.4.5. We only need to analyze a single step of our algorithm.

Lemma 5.2.9. *A single time step iteration of Algorithm 5.1 satisfies ε_0-differential privacy.*

Proof. The exponential mechanism as defined satisfies $\varepsilon_0/2$-differential privacy [MT]. On the other hand, $d_t(f_t)$ has sensitivity at most $1/n$ (the statistical query $f(x)$ minus a public quantity $f(x_t)$). Hence, $\widehat{d_t}(f_t)$ satisfies $\varepsilon_0/2$-differential privacy as well. The claim now follows from Theorem 2.4.4. ∎

Corollary 5.2.10 (Privacy). *Algorithm 5.1 satisfies*

1. *$(\varepsilon_0 T)$-differential privacy,*
2. *$(\varepsilon_0\sqrt{2T\log(1/\delta)} + T\varepsilon_0(e^{\varepsilon_0} - 1), \delta)$-differential privacy.*

5.2.4 Minimizing error while maintaining privacy

In this section we give two theorems that are each obtained directly from our previous analysis by minimizing the error of Algorithm 5.1, while maintaining either ε-differential privacy or (ε, δ)-differential privacy. In each case, the required setting of the parameter T and the privacy parameter is omitted from the theorem and instead stated explicitly only in the proof of each theorem.

Theorem 5.2.11 (implies Theorem 5.1.1). *Let $\varepsilon > 0$. Given a data set of size n over a universe $\mathcal{U}$ of size N and a set of statistical queries $\mathcal{Q}$, Algorithm 5.1 can be used to produce synthetic data x^* satisfying $(\varepsilon, 0)$-differential privacy and*

$$\mathbb{E}\,\mathrm{err}(x^*, \mathcal{Q}) \leqslant O\left(\frac{\log N \log k}{\varepsilon n}\right)^{1/3}.$$

Proof. To achieve ε-differential privacy over all we will run our algorithm with $\varepsilon_0 = \varepsilon/T$ and invoke Corollary 5.2.10. Now, let

$$\alpha(T) = \sqrt{\frac{\log N}{T}} + \frac{T\log k}{\varepsilon n}.$$

By Lemma 5.2.1, $\mathbb{E}\,\mathrm{err}(x^*, \mathcal{Q}) \leqslant O(\alpha(T))$. We would therefore like to minimize $\alpha(T)$ as a function of T. Writing $\alpha(T) = aT^{-1/2} + bT$ and taking the derivative, we see that the minimum obtains at $T^* = (a/2b)^{2/3}$, giving $\alpha^* = \alpha(T^*) =$

$2a^{2/3}b^{1/3}$, which corresponds to

$$T^* = \left(\frac{\varepsilon n\sqrt{\log N}}{2\log k}\right)^{2/3}, \qquad \alpha^* = 2\left(\frac{\log N \log k}{\varepsilon n}\right)^{1/3}.$$

Running Algorithm 5.1 with this setting of T and ε, hence gives the desired result. ∎

We can obtain a stronger error bound by passing from ε-differential privacy to (ε, δ)-differential privacy.

Theorem 5.2.12 (implies Theorem 5.1.2). *Let $\varepsilon > 0, \delta > 0$. Given a data set of size n over a universe $\mathcal{U}$ of size N and a set of statistical queries $\mathcal{Q}$, Algorithm 5.1 produces synthetic data x^* satisfying (ε, δ)-differential privacy and*

$$\mathbb{E}\,\mathrm{err}(x^*, \mathcal{Q}) \leqslant O\left(\frac{\sqrt{\log N \log(1/\delta)} \cdot \log k}{\varepsilon n}\right)^{1/2}.$$

Proof. To achieve (ε, δ)-differential privacy overall, by Corollary 5.2.10, we will run Algorithm 5.1 with privacy parameter ε_0 satisfying $\varepsilon = \varepsilon_0\sqrt{2T\log 1/\delta} + T\varepsilon_0(e^{\varepsilon_0} - 1)$.

Choosing $\varepsilon_0 = \varepsilon / C\sqrt{T\log(1/\delta)}$ for some constant $C > 0$ is sufficient. Let

$$\alpha(T) = \sqrt{\frac{\log N}{T}} + \frac{\sqrt{T\log(1/\delta)}\log k}{\varepsilon n}.$$

By Lemma 5.2.1, $\mathbb{E}\,\mathrm{err}(x^*, \mathcal{Q}) \leqslant O(\alpha(T))$. Again, we would like to minimize $\alpha(T)$ as a function of T. This is achieved for

$$T^* = \Theta\left(\frac{\sqrt{\log N}\varepsilon n}{\sqrt{\log(1/\delta)}\log k}\right),$$

giving $\alpha(T^*)$ that matches the bound stated in the theorem. ∎

5.3 Implementation and experimentation

In this section we consider an application of our general framework to the problem of contingency table release. We choose this particular problem because it exhibits interesting correlations between queries, as well as having a significant role in the practice of official statistics.

A contingency table reflects a set of k discrete attributes, where each record in the table has a setting of each attribute. A contingency table is commonly

represented by enumerating the list of all possible settings of the attributes and reporting the counts of the number of records with the associated setting. We can also do the same for a subset of the attributes, reporting the counts for each possible setting of the attributes in the subset, which is referred to as a *marginal*.

When statistical inference is performed over contingency tables, statisticians seek sets of *low-order* marginals, those containing relatively few attributes at a time, that explain the data well. Our goal, in releasing contingency tables, is to release data so that these low-order marginals are accurately preserved.

In previous work, Barak et al. [BCD+] describe an approach to differentially private contingency table release through the Fourier transformation. If we view a contingency table as vector, coordinates ordered lexicographically by [binary] attribute settings, the Fourier transformation corresponds to multiplication by the Hadamard matrix, defined recursively as

$$H_{n+1} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}, \qquad H_1 = [1].$$

Each result in the multiplication corresponds to the measurement of a Fourier coefficient. Intuitively, there is one such measurement for each subset of attributes, reflecting the counts of records with even and odd parities of attribute values among the subset of attributes.

These coefficients are interesting in that all low order marginals can be exactly recovered by examination of relatively few entries in the transformed vector, the measurements corresponding to subsets of size at most k. Rather than explain the production of marginals from these Fourier measurements (details can be found in Barak et al. [BCD+]) we simply exploit the connection by considering the corresponding count queries, and insisting on producing a distribution that respects them. The marginals can then be derived directly from the distribution.[2]

5.3.1 Experimental Set-up

For our experiments, we consider several data sets used in the statistical literature. The data sets range from relatively small (70 records) to substantial (21k records). We have avoided enormous data sets as they seem to occur less frequently in practice (truly large data sets are invariably segmented

[2]There is nothing wrong with explicitly using the Fourier transform to return to the marginals, but it is exciting to note that we do not need to specify the relationship between the measurements we take and the quantities of interest; we only need the relationships to exist. This is helpful when the dependence is complicated and/or inexact.

into subpopulations before analysis), and are not especially good indicators of algorithm performance (even the simplest algorithm works well). The challenge with differential privacy is getting it to work on smaller data sets, rather than larger. The data sets we consider are detailed in Table 5.1. There

	records	attributes	non-zero / total cells
mildew	70	6	22 / 64
czech	1841	6	63 / 64
rochdale	665	8	91 / 256
nltcs	21574	16	3152 / 65536

Table 5.1: Structural details of the four data sets we consider.

are several ways to measure the quality of our approaches. One that we will focus on is relative entropy, or KL divergence. This measurement has appealing properties for statistical inference, and is used in previous statistical work on the problem. Other measurements, for example directly measuring the error in marginal tables, certainly exist, but our goal is ultimately learning models that fit the data well, and it is not immediately clear how accuracy in these measurements result in statistical quality of fit.

Our experiments are intended both to compare our approach to the prior work of Barak et al. [BCD+] as well as to evaluate it in absolute terms. For the purposes of our experiments, Barak et al. will simply be represented by the approach that takes all low order Fourier measurements with a uniform level of accuracy; their approach involved an additional linear programming step, which we found hurts its performance with respect to relative entropy. For the absolute comparison, we invoke the work of Fienberg et al. [FRY] on several of these data sets where they report absolute numbers for quality of fit (in terms of relative entropy) without privacy constraints.

All of our experiments are done with ε-differential privacy, that is, $\delta = 0$. The absolute numbers improve in the privacy-utility trade-off if we permit a non-zero δ. However, the relationships between the curves can change; the improvement in ε one would see with a non-zero δ depends on the technique and the data set in a way we have not measured.

5.3.2 Improvements

We now consider several variations on the simple approach presented in Section 5.2 that can lead to noticeably improved performance. Although the worst case bounds do not improve, there is theoretical motivation for each of the improvements, which we also detail.

Iterating the Update On each iteration we select a query to measure based on the amount of error exhibited between our approximating distribution and the true data. The selected query is measured, and corrected. However, over the course of the algorithm, measurements may drift again. There is no privacy cost to re-processing a previous measurement, so we can take advantage of this to further decrease our potential function without re-interrogating the data.

Initialization Our potential function starts at the logarithm of the universe size, because our best guess at the outset is of a uniform distribution. This can sometimes be improved by taking a histogram of the values; by simply counting (with noise) the number of occurrences of each type of record, we can identify values that occur with substantial frequency and update the prior accordingly. This works well if there are several values with high frequency (as they contribute most to the potential function) but it does consume from the privacy budget, and reduces the accuracy allowed in the query measurement stage.

Adapting the Number of Rounds The number of rounds to conduct is an important parameter. Setting it too low results in not enough information extracted about the data, but setting it too high causes each round to give very noisy measurements, of little value. Instead, we can set the number adaptively, by starting with a very small epsilon value and asking queries until the observed signal drops below noise levels. At this point, if privacy budget still remains, we double epsilon and restart. As epsilon increases we will only drill deeper, each round asking at least as many questions as the last at twice the privacy cost, causing the cumulative cost to telescope and be within a factor of two of the final cost.

5.3.3 Small Datasets

We first evaluated our approach on several small data sets in common use by statisticians. Our findings here were fairly uniform across the data sets: the ability to measure only those queries that are informative about the data set results in substantial savings over taking all possible measurements. We evaluated both our theoretically pure algorithm and its heuristic improvement as discussed in the previous section, against a modified version of the algorithm of Barak et al. [BCD+] (integrating the multiplicative weights of Hardt-Rothblum [HR]), and the accepted "good" non-private relative entropy values from Fienberg et al. [FRY]. The trade-off between relative entropy and ε for three data sets appears in Figure 5.2. In each case, we see that we

noticeably improve on the algorithm of Barak et al., and in many cases our heuristic approach matches the good non-private values.

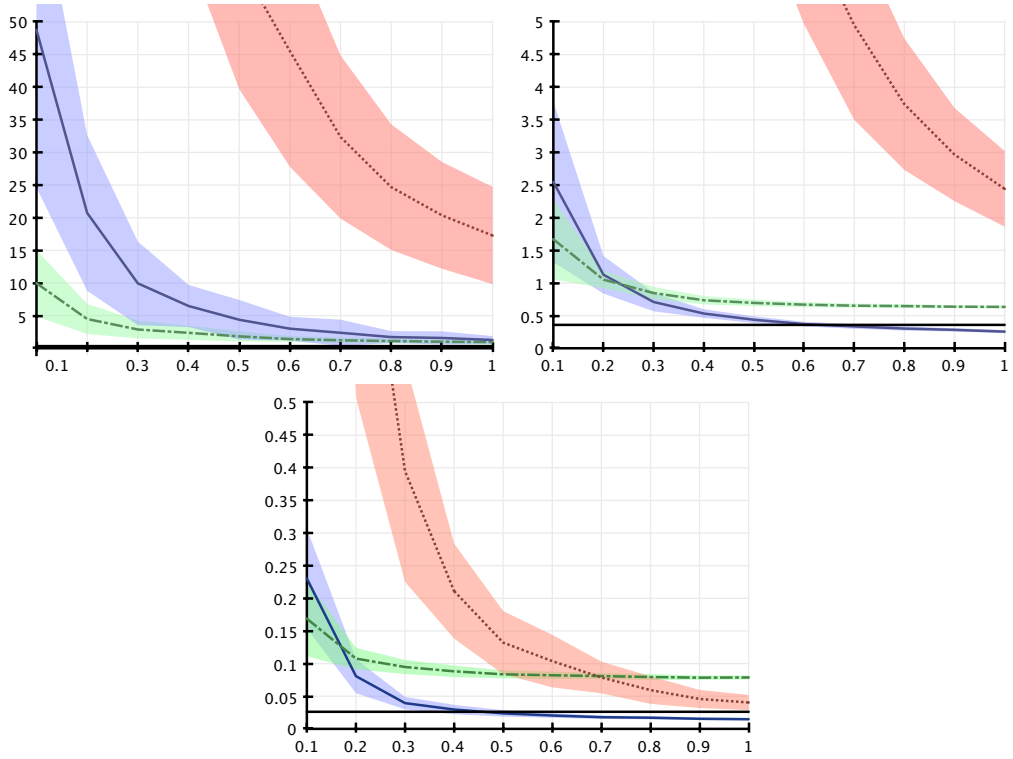

Figure 5.2: Curves describing the behavior of algorithms on the mildew, rochdale, and czech data sets, respectively. The x-axis is the value of epsilon guaranteed, and the y-axis is the relative entropy between the produced distribution and actual data set. The lines represent averages across 100 runs, and the corresponding shaded areas one standard deviation in each direction. Red (dashed) represents the modified Barak et al. algorithm, green (dot-dashed) represents the unoptimized use of the exponential mechanism to select queries for the multiplicative weights algorithm, and blue (solid) represents the optimized version thereof. The solid black horizontal line is the stated relative entropy values from Fienberg et al.

5.3.4 Large Dataset

We also consider a larger data set, the National Long-Term Care Study (NLTCS) in Figure 5.3. This data set contains orders of magnitudes more records, and has 16 binary attributes. For our initial settings, maintaining all three-way Fourier measurements, we see similar behavior as above: the ability to choose the measurements that are important allows substantially higher accuracy on those that matter.

However, we see that the algorithm of Barak et al. [BCD+] is substantially more competitive in the regime where we are interested in querying all two-way marginals, rather than the default three we have been using. In this case, for values of epsilon at least 0.1, it seems that there is enough signal present to simply measure all such Fourier coefficients; each is sufficiently informative that measuring substantially fewer at higher accuracy imparts less information, rather than more.

For every data set and query set, there is some sufficiently high epsilon level where the judicious selection of queries is no longer required. In such regimes, the approach we present in this chapter does not provide an improvement over more naive approaches. The impact of our approach returns if we increase the order of marginal that must be preserved (dramatically increasing the number of measurements Barak et al. would take) or if we decrease epsilon to a level such that the majority of two-way Fourier coefficients are not above the noise level. However, the analyst's goal should be to get the right output for the analysis task at hand, under the supplied privacy constraints. In some cases this may not require the use of our advanced query selection.

5.4 Conclusions

We have studied a simple algorithm for releasing data maintaining a high fidelity to the protected source data, as well as differential privacy with respect to the records. The approach builds upon the multiplicative weights approach of [HR], by introducing the exponential mechanism [MT] as a more judicious approach to determining which measurements to take. The theoretical analysis matches the state of the art, and experimentally we have evidence that for many interesting parameters it represents a substantial improvement over existing techniques.

As well as matching the best known theoretical bounds and improving experimental performance, the algorithm is both simple to implement and simple to use. An analyst does not require a complicated mathematical understanding of the nature of the queries (as the community has for Fourier coefficients and marginal tables), but rather only needs to enumerate those measurements that should be preserved. We hope that this generality leads to a broader class of high fidelity differentially-private data releases for a variety of data domains.

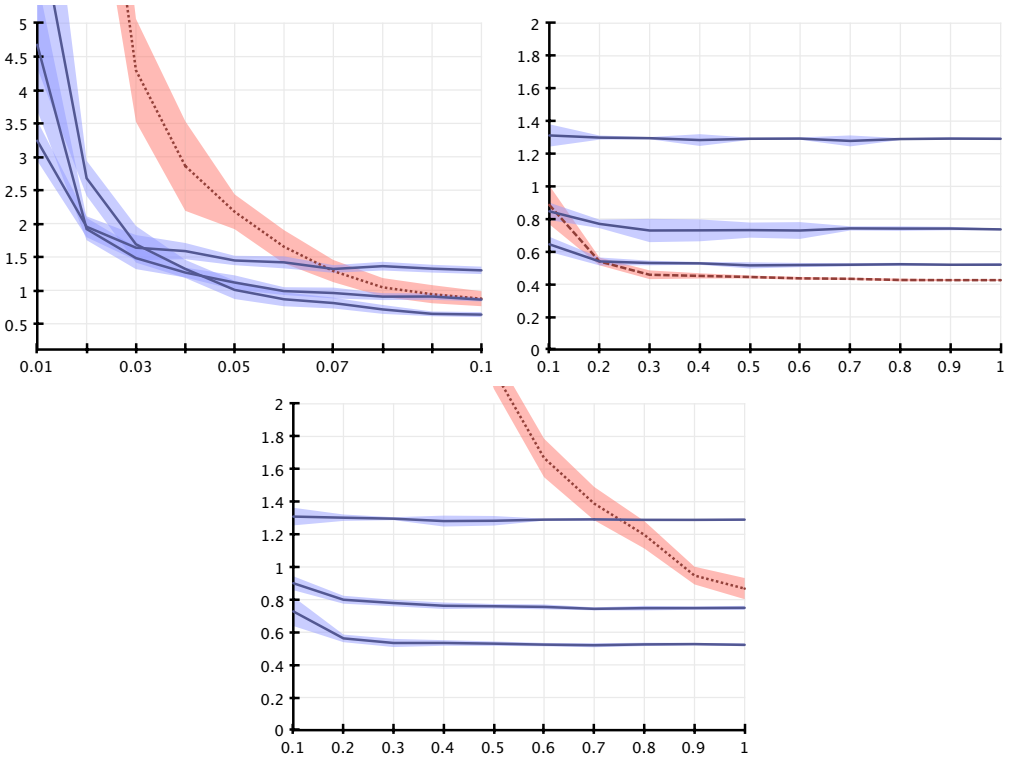

Figure 5.3: Curves comparing our approach with that of Barak et al on the National Long Term Care Survey. The red (dashed) curve represents Barak et al, and the multiple blue (solid) curves represent multiplicative weights combined with the exponential mechanism, with 20, 30, and 40 queries (top to bottom, respectively). From left to right, the first two figures correspond to degree 2 Fourier coefficients, and the third to degree 3 Fourier coefficients. We see that the exponential mechanism improves accuracy when the privacy requirements are strong relative to the number of measurements (the first and third graphs). The middle graph demonstrates that with few measurements and sufficient privacy budget, one does best by simply measuring everything. As before, the x-axis is the value of epsilon guaranteed, and the y-axis is the relative entropy between the produced distribution and actual data set. The lines represent averages across only 10 runs, owing to the high complexity of Barak et al on this many-attributed data set, and the corresponding shaded areas one standard deviation in each direction.

Chapter 6

Releasing Conjunctions and the Statistical Query Barrier

One of the most important classes of statistical queries on the data set are Boolean conjunctions which we introduced in Section 2.7.1. Recall that a boolean conjunction corresponding to a subset $S \subseteq [d]$ counts what fraction of the individuals have each attribute in S set to 1. A major open problem in privacy-preserving data analysis is to efficiently create a differentially private synopsis of the data set that accurately encodes answers to all Boolean conjunctions. In this chapter we see an algorithm with running time polynomial in d, which outputs a differentially private data structure that represents 99% of all Boolean conjunctions up to an additive error of 1%.

Our result is more general and applies to any collection of queries that can be described by a low sensitivity *submodular* function. Submodularity is a property that often arises in data analysis and machine learning problems, including in problems for which privacy is a first-order design constraint[1]. Imagine, for example, a social network on d vertices. A data analyst may wish to analyze the size of the cuts induced by various subsets of the vertices. Here, our result provides a data structure that represents all cuts up to a small average error. Another important example of submodularity is the *set-coverage* function, which given a set system over elements in some universe U, represents the number of elements that are covered by the union of any collection of the sets.

The size of our data structure grows exponentially in the inverse error desired, and hence we can represent submodular functions only up to constant error if we want polynomial complexity. *Can any efficient algorithm do even better?* We give evidence that in order to do better, fundamentally new techniques are needed. Specifically, we show that no polynomial-time algorithm can do substantially better if the algorithm permits an implementation that only accesses the database through statistical queries (cf. Section 2.9.1),

[1] For example, Kempe, Kleinberg, and Tardos show that for two common models of influence propagation on social networks, the function capturing the "influence" of a set of users (perhaps the targets of a viral marketing campaign) is a monotone submodular function [KKT].

regardless of whether such an implementation is privacy-preserving.

How do we show this? First, putting aside privacy concerns, we pose the following question: *How many statistical queries to a data set are necessary and sufficient in order to approximately answer* all *queries in a class C?* We show that the number of statistical queries necessary and sufficient for this task is, up to a factor of $O(d)$, equal to the agnostic learning complexity of C (over arbitrary distributions) in Kearns' statistical query (SQ) model [Kea]. Now, using an SQ lower bound for agnostically learning monotone conjunctions shown by Feldman [Fel], this connection implies that no polynomial-time algorithm operating in the SQ-model can release even monotone conjunctions to subconstant error. (Since releasing submodular functions is even more general, the lower bound carries over to that setting as well.)

While the characterization above is independent of privacy concerns, it has two immediate implications for private data release:

— Firstly, it also characterizes what can be released in the *local privacy* model of Kasiviswanathan et al. [KLN+]; this follows from the fact that [KLN+] showed that SQ algorithms are precisely what can be computed in the local privacy model.

— Secondly, and perhaps even more importantly, it gives us the claimed unconditional lower bounds on the running time of any query-release algorithm that permits an implementation using only statistical queries—regardless of whether its privacy analysis can be carried out in the local privacy model. To our knowledge, this class includes almost all privacy preserving algorithms developed to date, including the Median Mechanism of [RR] and the multiplicative weights method we saw in Chapter 4.[2] Note that these mechanisms cannot be implemented in the local privacy model while preserving their privacy guarantees, because they will have to make too many queries. Indeed, they are capable of releasing conjunctions to subconstant error! Yet, they can be implemented using only statistical queries, and so our lower bounds apply to their running time.

To summarize, our results imply that if we want to develop efficient algorithms to solve the query release problem for classes as expressive as monotone conjunctions, we need to develop techniques that are able to sidestep this *statistical query barrier*. On a conceptual note, our results present new reductions from problems in differential privacy to problems in learning theory.

[2] A notable exception is the private parity-learning algorithm of [KLN+], which explicitly escapes the statistical query model.

6.1 Main results

In this section we give an informal statement of our theorems with pointers to the relevant sections. Our theorem on approximating submodular functions is proved in Section 6.3. The definition of submodularity is found in the Preliminaries (Section 6.2).

Informal Theorem 6.1.1 (Approximating submodular functions). *Let $\alpha > 0, \beta > 0$. Let $f : \{0,1\}^d \to [0,1]$ be a submodular function. Then, there is an algorithm with runtime $d^{O(\log(1/\beta)/\alpha^2)}$ which produces an approximation $h : \{0,1\}^d \to [0,1]$ such that $\mathbb{P}_{x \in \{0,1\}^d}\{|f(x) - h(x)| \leqslant \alpha\} \geqslant 1 - \beta$. The algorithm is allowed to make oracle queries to f at arbitrary points in $\{0,1\}^d$.*

In Section 6.4 we then show how this algorithm gives the following differentially private release mechanism for Boolean conjunctions. The definition of differential privacy is given in Section 6.2.

Informal Theorem 6.1.2 (Differentially private query release for conjunctions). *Let $\alpha > 0, \beta > 0$. There is an ε-differentially private algorithm with runtime $d^{O(\log(1/\beta)/\alpha^2)}$ which releases the set of Boolean conjunctions with error at most α on a $1 - \beta$ fraction of the queries provided that $|D| \geqslant d^{O(\log(1/\beta)/\alpha^2)}/\varepsilon$.*

The guarantee in our theorem can be refined to give an α-approximation to a $1 - \beta$ fraction of the set of w-way conjunctions (conjunctions of width w) for all $w \in \{1, \ldots, d\}$. Nevertheless, our algorithm has the property that the error may be larger than α on a small fraction of the queries. We note, however, that for $\beta \leqslant \alpha^p/2$ our guarantee is stronger than error α in the ℓ_p-norm which is also a natural objective. Recall that we worked with ℓ_2-error in Chapter 3. From a practical point of view, it also turns out that some privacy-preserving algorithms in the literature indeed only require the ability to answer *random* conjunction queries privately, e.g., [JPW].

Finally, in Section 6.5, we study the general query release problem and relate it to the agnostic learning complexity in the Statistical Query model.

Informal Theorem 6.1.3 (Equivalence between query release and agnostic learning). *Suppose there exists an algorithm that learns a class C up to error α under arbitrary distributions using at most q statistical queries. Then, there is a release mechanism for C that makes at most $O(qd/\alpha^2)$ statistical queries.*

Moreover, any release mechanism for C that makes at most $2q$ statistical queries implies an agnostic learner that makes at most q queries.

While both reductions preserve the query complexity of the problem neither reduction preserves runtime. We also note that our equivalence characterization is more general than what we stated: the same proof shows that

agnostic learning of a class C is (up to small factors) information theoretically equivalent to releasing the answers to all queries in a class C for any class of algorithms that may access the database only in some restricted manner. The ability to make only SQ queries is one restriction, and the requirement to be differentially private is another. Thus, we also show that on a class by class basis, the privacy cost of releasing the answers to a class of queries using any technique is not much larger than the privacy cost of simply optimizing over the same class to find the query with the highest value, and vice versa.

Our techniques. Our algorithm is based on a structural theorem about general submodular functions $f : 2^U \to [0,1]$ that may be of independent interest. Informally, we show that any submodular function has a "small" "approximate" representation. Specifically, we show that for any $\alpha > 0$, there exist at most $|U|^{2/\alpha}$ submodular functions g_i such that each g_i satisfies a strong Lipschitz condition, and for each $S \subset U$, there exists an i such that $f(S) = g_i(S)$. We then take advantage of Vondrak's observation that Lipschitz-continuous submodular functions are *self-bounding*, which allows us to apply recent dimension-free concentration bounds for self-bounding functions [Von]. These concentration results imply that if we associate each function g_i with its expectation, and respond to queries $f(S)$ with $\mathbb{E}[g_i(S)]$ for the appropriate g_i, then most queries are answered to within only α additive error. This yields an algorithm for *learning* submodular functions over product distributions, which can easily be made privacy preserving.

Our characterization of the query complexity of the release problem in the SQ model uses the multiplicative weights method [LW, AHK] similar to how it is used in Chapter 5. That is we maintain a distribution over the universe on which the queries are defined. What is new is the observation that an agnostic learning algorithm for a class C can be used to find a query from C that distinguishes between the true data set and our distribution as much as possible. Such a query can then be used in the multiplicative weights update to reduce the relative entropy between the true data set and our distribution significantly. Since the relative entropy is nonnegative there can only be a few such steps before we find a distribution which provides a good approximation to the true data set on *all* queries in the class C.

6.1.1 Related work on learning submodular functions

In this section we discuss previous work on learning submodular functions since it was not part of Chapter 2.

The problem of learning submodular functions was recently introduced by Balcan and Harvey [BH]; their PAC-style definition was different from

previously studied point-wise learning approaches [GHIM, SF]. For product distributions, Balcan and Harvey give an algorithm for learning monotone, Lipschitz continuous submodular functions up to constant *multiplicative* error using only random examples. [BH] also give strong lower bounds and matching algorithmic results for non-product distributions. Our main algorithmic result is similar in spirit, and is inspired by their concentration-of-measure approach. Our model is different from theirs, which makes our results incomparable. We introduce a decomposition that allows us to learn arbitrary (i.e. potentially non-Lipschitz, non-monotone) submodular functions to constant *additive* error. Moreover, our decomposition makes value queries to the submodular function, which are prohibited in the model studied by [BH].

6.2 Preliminaries

In this chapter, a *counting query* is specified by a predicate $q\colon X \to [0,1]$. We will denote the answer to a count query (with some abuse of notation) by $q(D) = \frac{1}{n}\sum_{x\in D} q(X)$. Note that a count query can differ by at most $1/n$ on any two adjacent databases. In particular, adding Laplacian noise of magnitude $1/\varepsilon n$, denoted $Lap(1/\varepsilon n)$, guarantees ε-differential privacy on a single count query (see Chapter 2 for details).

We will state our algorithms in Kearns' statistical query (SQ) model as introduced in Section 2.9.1.

Query release. A *concept class* (or *query class*) is a distribution over *concepts* (or predicates) from $X \to [0,1]$, e.g., the uniform distribution over a finite set of predicates.

Definition 6.2.1 (Query Release)**.** Let C be a concept class. We say that an algorithm A (α,β)-*releases* C over a data set D if $\mathbb{P}_{q\sim C}\{|q(D)-A(q)| \leqslant \alpha\} \geqslant 1-\beta$.

Specifically, we are interested in algorithms which release C using few statistical queries to the underlying data set. We will study the query release problem by considering the function $f(q) = q(D)$. In this setting, releasing a concept class C is equivalent to *approximating* the function q is the following sense

Definition 6.2.2. We say that an algorithm A (α,β)-approximates a function $f\colon 2^U \to [0,1]$ over a distribution P if $\mathbb{P}_{S\sim P}\{|f(S)-A(S)| \leqslant \alpha\} \geqslant 1-\beta$.

For many concept classes of interest, the function $f(q)$ will be *submodular*, defined next.

Submodularity. Given a universe U, a function $f : 2^U \to \mathbb{R}$ is called *submodular* if for all $S, T \subset U$ it holds that $f(S \cup T) + f(S \cap T) \leqslant f(S) + f(T)$. We define the *marginal value* of x (or *discrete derivative*) at S as $\partial_x f(S) = f(S \cup \{x\}) - f(S)$.

Fact 6.2.3. *A function f is submodular if and only if $\partial_x f(S) \geqslant \partial_x f(T)$ for all $S \subseteq T \subseteq U$ and all $x \in U$.*

Definition 6.2.4. A function $f : 2^U \to \mathbb{R}$ is ρ-Lipschitz if for every $S \subseteq U$ and $x \in U$, $|\partial_x f(S)| \leqslant \rho$.

Concentration bounds for submodular functions. The next lemma was shown by Vondrak [Von] building on concentration bounds for so-called self-bounding functions due to [BLM1, BLM2].

Lemma 6.2.5 (Concentration for submodular functions). *Let $f : 2^U \to \mathbb{R}$ be a 1-Lipschitz submodular function. Then for any product distribution $\mathcal{P}$ over 2^U, we have*

$$\mathop{\mathbb{P}}_{S \sim \mathcal{P}} \{|f(S) - \mathbb{E} f(S)| \geqslant t\} \leqslant 2 \exp\left(-\frac{t^2}{2(\mathbb{E} f(S) + 5t/6)}\right), \tag{6.1}$$

where the expectations are taken over $S \sim \mathcal{P}$.

We obtain as a simple corollary

Corollary 6.2.6. *Let $f : 2^U \to [0,1]$ be a ρ-Lipschitz submodular function. Then for any product distribution $\mathcal{P}$ over 2^U, we have*

$$\mathop{\mathbb{P}}_{S \sim \mathcal{P}} \{|f(S) - \mathbb{E} f(S)| \geqslant \rho t\} \leqslant 2 \exp\left(-\frac{t^2}{2(1/\rho + 5t/6)}\right), \tag{6.2}$$

where the expectations are taken over $S \sim \mathcal{P}$.

6.3 Approximating submodular functions

Our algorithm for approximating submodular functions is based on a structural theorem, together with some strong concentration inequalities for submodular functions (see Lemma 6.2.5). In this section, we prove our structure theorem, present our algorithm, and prove its correctness.

6.3.1 Monotone submodular functions

We begin with a simpler version of the structure theorem. This version will be sufficient for approximating bounded monotone submodular functions from value queries, and will be the main building block in our stronger results,

which will allow us to approximate arbitrary bounded submodular functions, even from "tolerant" value queries.

Our structure theorem follows from an algorithm that decomposes a given submodular function into Lipschitz submodular functions. The algorithm is presented next and analyzed in Lemma 6.3.1.

Decompose(f, ρ):
Input: Query access to a monotone submodular function $f\colon 2^U \to [0,1]$ and parameter $\rho \geqslant 0$.

1. Let $\prec$ denote an arbitrary ordering of U.
2. Let $\mathcal{I} \leftarrow \{\emptyset\}$
3. For each $x \in U$ (in ascending order under $\prec$):
   (a) Let $\mathcal{I}' \leftarrow \emptyset$
   (b) For each $B \in \mathcal{I}$, if $\partial_x f(B) > \rho$, then update $\mathcal{I}' \leftarrow \mathcal{I}' \cup \{B \cup \{x\}\}$
   (c) Let $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{I}'$
4. Let $V(S) = \{x \in U \mid \partial_x f(S) \leqslant \rho\}$ denote the set of elements that have small marginal value with respect to $S \subseteq U$.

Output: the collection of functions $\mathcal{G} = \{g^B \mid B \in \mathcal{I}\}$, where for $B \in \mathcal{I}$ we define the function $g^B\colon 2^{V(B)} \to [0,1]$ as $g^B(S) = f(S \cup B)$.

Figure 6.1: Decomposition for submodular functions (Algorithm 6.1)

Lemma 6.3.1. *Given any submodular function $f\colon 2^U \to [0,1]$ and $\rho \geqslant 0$, Algorithm 6.1 makes the following guarantee. There are maps $F, T\colon 2^U \to 2^U$ such that:*

1. (Lipschitz) *For every $g^B \in \mathcal{G}$, g^B is submodular and satisfies $\sup_{x \in V(B), S \subseteq V(B)} \partial_x g^B(S) \leqslant \rho$.*
2. (Completeness) *For every $S \subseteq U$, $F(S) \subseteq S \subseteq V(F(S))$ and $g^{F(S)}(S) = f(S)$.*
3. (Uniqueness) *For every $S \subseteq U$ and every $B \in \mathcal{I}$, we have $F(S) = B$ if and only if $B \subseteq S \subseteq V(B)$ and $S \cap T(B) = \emptyset$.*
4. (Size) *The size of $\mathcal{G}$ is at most $|\mathcal{G}| = |U|^{O(1/\rho)}$. Moreover, given oracle access to f, we compute F, V, T in time $|U|^{O(1/\rho)}$.*

Note that the lemma applies to non-monotone submodular functions f as well; however, since our release algorithm will require the stronger condition $\sup_{x \in V(B), S \subseteq V(B)} |\partial_x g(S)| \leqslant \rho$, the lemma will only be sufficient for releasing monotone submodular functions (where it holds that $|\partial_x g(S)| \leqslant \rho \iff \partial_x g(S) \leqslant \rho$). We will return to the non-monotone case later.

Proof. Algorithm 6.1 always terminates and we have the following bound on the size of $\mathcal{I}$.

Claim 6.3.2. $|\mathcal{I}| \leqslant |U|^{1/\rho}$

Proof. Let $B \in \mathcal{I}$ be a set, $B = \{x_1, \ldots, x_{|B|}\}$. Let $B_0 = \emptyset$ and $B_i = \{x_1, \ldots, x_i\}$ for $i = 1, \ldots, |B| - 1$. Then

$$1 \geqslant f(B) = \sum_{i=0}^{|B|-1} \partial_{x_{i+1}} f(B_i) > |B| \cdot \rho. \tag{6.3}$$

Therefore, it must be that $|B| \leqslant 1/\rho$, and there are at most $|U|^{1/\rho}$ such sets over $|U|$ elements. ∎

Item 1 is shown next.

Claim 6.3.3 (Lipschitz). *For every $g^B \in \mathcal{G}$, g^B is submodular and $\sup_{x \in V(B), S \subseteq V(B)} \partial_x g^B(S) \leqslant \rho$.*

Proof. Submodularity follows from the fact that g^B is a "shifted" version of f. Specifically, if $T \subseteq S$, then $\partial_x g^B(S) = \partial_x f(B \cup S) \leqslant \partial_x f(B \cup T) = \partial_x g^B(T)$, where the inequality is by submodularity of f.

To establish the Lipschitz property, we note that by the definition of V, $\partial_x f(B) \leqslant \rho$ for every $x \in V(B)$. Also, by the submodularity of f, we have $\partial_x g^B(S) = \partial_x f(B \cup S) \leqslant \partial_x f(B) \leqslant \rho$. ∎

Definition of F and proof of Item 2. Now we turn to constructing the promised mappings F and T in order to Properties 2 and 3. Roughly, we want $F(S)$ to choose a maximal set in $\mathcal{I}$ such that $F(S) \subseteq S$, in order to assure that $S \subseteq V(F(S))$. This task is complicated by the fact that there could be many such sets. We want to be able to choose a unique such set, and moreover, given any such set B, determine efficiently if $F(S) = B$. To achieve the former task, we define a specific, deterministic mapping $F(S)$ and to achieve the latter we will carefully define the mapping T.

We define $F(S)$ as follows:

let $j \leftarrow 0$, $B_j \leftarrow \emptyset$
for $x \in U$ (in ascending order under $\prec$) **do**

if $x \notin V(B_j)$ and $x \in S$ **then** $B_{j+1} \leftarrow B_j \cup \{x\}$, $j \leftarrow j+1$
end for
return $F(S) = B_j$.

Note that this procedure is similar to the procedure we use to construct $\mathcal{I}$. To construct $\mathcal{I}$, we gradually constructed a tree of sets, where each set $B \in \mathcal{I}$ had a child for every set $B \cup \{x\}$ such that x has high influence on B ($x \notin V(B)$). The procedure $F(S)$ differs in that it only constructs a single root-leaf path in this tree, where for each B_j in the path, the next set in the path is $B_j \cup \{x\}$ where x is the *minimal* $x \in S$ that has high influence on B_j (and has not already been considered by $F(S)$. We will use $P(S) = (B_0 \subset B_1 \subset \cdots \subset F(S))$ to denote this path, which is the sequence of intermediate sets B_j in the execution of $F(S)$. Given these observations, we can state the following useful facts about F.

Fact 6.3.4. *If $F(S) = B$, then $P(S) = P(B)$. Moreover, for every $S \in U$, $P(S) \subseteq \mathcal{I}$.*

We can now establish Property 2 by the following claim.

Claim 6.3.5 (Completeness). *For every $S \subseteq U$, $F(S) \subseteq S \subseteq V(F(S))$, and $g^{F(S)}(S) = f(S)$.*

Proof. Let $P(S) = B_0 \subset B_1 \subset \cdots \subset F(S)$. $F(S)$ always checks that $x \in S$ before including an element x, so $F(S) \subseteq S$. To see that $S \subseteq V(F(S))$, assume there exists $x \in S \setminus V(F(S))$. By submodularity we have $\partial_x f(B_j) \geqslant \partial_x f(F(S)) > \rho$ for every set B_j. But if $\partial_x f(B_j) > \rho$ for every B_j and $x \in S$, it must be that $x \in F(S)$. But then $\partial_x f(F(S)) = 0$, contradicting the fact that $x \notin V(F(S))$.

Finally, we note that since $S \subseteq V(F(S))$, $g^{F(S)}(S)$ is defined (S is in the domain of $g^{F(S)}$) and since $F(S) \subseteq S$, $g^{F(S)}(S) = f(F(S) \cup S) = f(S)$. ∎

Definition of T and proof of Item 3. We will now define the mapping T. The idea is to consider a set $B \in \mathcal{I}$ and $P(B)$ and consider all the elements we had to "reject" on the way from the root to B. We say that an element $x \in U$ is "rejected" if, when x is considered by $F(S)$, it has high influence on the current set, but is not in B. Since any set S such that $B = F(S)$ satisfies $P(S) = P(B)$ (Fact 6.3.4), and any set S that contains a rejected element would have taken a different path, we will get that the elements $x \in T(B)$ "witness" the fact that $B \neq F(S)$. We define the map $T(B)$ as follows:

let $j \leftarrow 0$, $B_j \leftarrow \emptyset$, $R \leftarrow \emptyset$
for $x \in U$ (in ascending order under $\prec$) **do**
  if $x \notin V(B_j)$ and $x \notin B$ **then** $R \leftarrow R \cup \{x\}$
  else if $x \notin V_{B_j}$ and $x \in B$ **then** $B_{j+1} \leftarrow B_j \cup \{x\}$, $j \leftarrow j+1$
end for
return $T(B) = R$.

We'll establish Property 3 via the following two claims.

Claim 6.3.6. *If $B = F(S)$, then $B \subseteq S \subseteq V(B)$ and $S \cap T(B) = \emptyset$.*

Proof. We have already demonstrated the first part of the claim in Claim 6.3.5, so we focus on the claim that $S \cap T(B) = \emptyset$. By Fact 6.3.4, every set S s.t. $B = F(S)$ satisfies $P(S) = P(B)$. Let $(B_0 \subset B_1 \subset \cdots \subset B) = P(B)$. Suppose there is an element $x \in S \cap T(B)$. Then there is a set B_j such that $x \notin V(B_j)$ and $x \notin B$. But since $x \notin V(B_j)$ and $x \in S$, it must be that $x \in B_{j+1}$, contradicting the fact that $B_{j+1} \subseteq B$. ■

Now we establish the converse.

Claim 6.3.7. *If $B \subseteq S \subseteq V(B)$, $S \cap T(B) = \emptyset$, then $B = F(S)$.*

Proof. Suppose for the sake of contradiction that there a set $B' \neq B$ such that $B' = F(S)$. There exists an element $x \in B \triangle B'$, and we consider the minimal such x under $\prec$. Let $P(B) = (B_0 \subset B_1 \subset \cdots \subset B)$ and $P(S) = P(B') = (B'_0 \subset B'_1 \subset \cdots \subset B')$. Since x is minimal in $B \triangle B'$, there must be j be such that $B_i = B'_i$ for all $i \leqslant j$, but $x \in B_{j+1} \triangle B'_{j+1}$. Consider two cases:

1. $B \supset B'$. Thus $x \in B \setminus B'$ Moreover, since $x \in B \subseteq S$, it must be that when x was considered in the execution of $F(S)$, and B'_j was the current set, it was the case that $x \in V(B'_j)$. But $B_j = B'_j$, so $x \in V(B_j)$, contradicting the fact that $x \in B_{j+1}$.

2. $B \not\supset B'$. Thus $x \in B' \setminus B$. Since $x \in B' = F(S) \subseteq S$ (Claim 6.3.5), we have $x \in S$. Moreover, since $x \in B'_{j+1}$ we must have $x \notin V(B'_j) = V(B_j)$. Thus we have $x \notin V(B_j)$ and $x \notin B$, which implies $x \in T(B)$, by construction. Thus $S \cap T(B) \neq \emptyset$, a contradiction.

■

The previous two claims establish Item 3.

Finally we observe that the enumeration of $\mathcal{I}$ requires time at most $|U| \cdot |\mathcal{I}| = |U|^{O(1/\rho)}$, since we iterate over each element of U and then iterate over each set currently in $\mathcal{I}$. We also note that we can compute the mappings F and T in time linear in $|\mathcal{I}| = |U|^{O(1/\rho)}$ and can compute $V(B)$ in time linear in $|U|$. These observations establish Property 4 and complete the proof of Lemma 6.3.1. ■

Lemma 6.3.8 (Lemma 6.3.1 with tolerance). *Given any submodular function $f : 2^U \to [0,1]$ and $\rho > 0$, Algorithm 6.2 makes the following guarantee. There are maps $F, T : 2^U \to 2^U$ satisfying Item 1—Item 4 of Lemma 6.3.1 and moreover, can be computed using tolerant queries to f with tolerance $\rho/12$.*

Input: Tolerant oracle access to a submodular function $f : 2^U \to [0,1]$ with tolerance at most $\rho/12$ and parameter $\rho > 0$.

Let $\widetilde{f}$ denote the function specified by tolerant oracle queries to f such that for every $S \subseteq U$, $|f(S) - \widetilde{f}(S)| \leqslant \rho/12$.
Let $\prec$ denote an arbitrary ordering of U.
Let $\mathcal{I} \leftarrow \{\emptyset\}$
for $x \in U$ (in ascending order under $\prec$) **do**
  $\mathcal{I}' \leftarrow \emptyset$
  for $B \in \mathcal{I}$ **do**
    if $\partial_x \widetilde{f}(B) > \rho/3$ **then** $\mathcal{I}' \leftarrow \mathcal{I}' \cup \{B \cup \{x\}\}$
  end for
  $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{I}'$
end for
Let $V(S) = \{x \in U \mid \partial_x \widetilde{f}(S) \leqslant 2\rho/3\}$ denote the set of elements that have small marginal value with respect to $S \subseteq U$.

Output: the collection of functions $\mathcal{G} = \{g^B \mid B \in \mathcal{I}\}$, where for $B \in \mathcal{I}$ we define the function $g^B : 2^{V(B)} \to [0,1]$ as $g^B(S) = f(S \cup B)$.

Figure 6.2: Decomposition for monotone submodular functions from tolerant queries

Proof. Throughout the proof, we will assume that the oracle always gives the same answer to each query. Thus the function $\widetilde{f}$ defined in Algorithm 6.2 is well defined. Note that $\widetilde{f}(S)$ need not be submodular even if f is, however, we can assume that we have exact oracle access to $\widetilde{f}(S)$. Also note that, since we can compute $\partial_x f(S)$ using two queries to f, we are guaranteed that for every $S \subseteq U$, and $x \in U$,

$$|\partial_x \widetilde{f}(S) - \partial_x f(S)| \leqslant \rho/6. \tag{6.4}$$

Observe that Algorithm 6.2 differs from Algorithm 6.1 only in the choice of parameters. The analysis required to establish the Lemma is also a natural modification of the analysis of Lemma 6.3.1, so we will refer the reader to the proof of that Lemma for several details and only call attention to the steps of the proof that require modification.

We will proceed by running through the construction of Lemma 6.3.1 on $\widetilde{f}(S)$ using $\rho/3$ as the error parameter. Since the argument is a fairly straightforward modification to Lemma 6.3.1, we will refer the reader to the proof of that Lemma for several details, and only call attention to the steps of the proof that require modification.

First, we establish a bound on the size of $\mathcal{I}$

Claim 6.3.9. $|\mathcal{I}| \leqslant |U|^{6/\rho}$

Proof. Let $B \in \mathcal{I}$ be a set, $B = \{x_1, \ldots, x_{|B|}\}$. Let $B_0 = \emptyset$ and $B_i = \{x_1, \ldots, x_i\}$ for $i = 1, \ldots, |B| - 1$. Then

$$1 \geqslant f(B) = \sum_{i=0}^{|B|-1} \partial_{x_{i+1}} f(B_i) \geqslant \sum_{i=0}^{|B|-1} \left(\partial_{x_{i+1}} \widetilde{f}(B_i) - \rho/6 \right) > |B| \cdot (\rho/3 - \rho/6) = |B| \cdot \rho/6 . \tag{6.5}$$

Therefore, it must be that $|B| \leqslant 6/\rho$, and there are at most $|U|^{6/\rho}$ such sets over $|U|$ elements. ■

Item 1 is shown next.

Claim 6.3.10 (Lipschitz). *For every $g^B \in \mathcal{G}$, g^B is submodular and*

$$\sup_{x \in V(B), S \subseteq V(B)} \partial_x g^B(S) \leqslant \rho .$$

Proof. The proof of submodularity is identical to Claim 6.3.3. To establish the Lipschitz property, observe that for every $B \subseteq U$, and every $x \in V(B)$, $\partial_x f(B) \leqslant \partial_x \widetilde{f}(B) + \rho/6 \leqslant \rho$. ■

Definition of F and proof of Item 2. In addition to the sets $V(B) = \{x \in U \mid \partial_x \widetilde{f}(B) \leqslant 2\rho/3\}$, we will define the sets $V'(B) = \{x \in U \mid \partial_x \widetilde{f}(B) \leqslant \rho/3\}$, note that for every $B \subseteq U$, $V'(B) \subseteq V(B)$. We define the promised mapping $F(S)$ in the the same manner as in the proof of Lemma 6.3.1, but we use V' in place of V to decide whether or not we select an element x for inclusion in the set $F(S)$.

Now we establish Item 2 via the following claim, analogous to Claim 6.3.5 in the proof of Lemma 6.3.1

Claim 6.3.11 (Completeness). *For every $S \subseteq U$, $F(S) \subseteq S \subseteq V(F(S))$. Moreover, $g^{F(S)}(S) = f(S)$.*

Proof. Let $P(S) = B_0 \subset B_1 \subset \cdots \subset F(S)$. The fact that $F(S) \subseteq S$ follows as in Claim 6.3.5. To see that $S \subseteq V(F(S))$, assume there exists $x \in S \setminus V(F(S))$. By submodularity of f, and (6.4), we have

$$\partial_x \widetilde{f}(B_j) \geqslant \partial_x f(B_j) - \rho/6 \geqslant \partial_x f(F(S)) - \rho/6 > \partial_x \widetilde{f}(F(S)) - \rho/3 > \rho/3.$$

Thus, $\partial_x \widetilde{f}(B_j) > \rho/3$ for every set B_j. But if $\partial_x f(B_j) > \rho/3$ for every B_j and $x \in S$, then $x \notin V'(B_j)$ for every B_j, and it must be that $x \in F(S)$. But then $\partial_x f(F(S)) = 0$, contradicting the fact that $x \notin V(F(S))$.

The fact that $g^{F(S)}(S) = f(S)$ follows as in the proof of Claim 6.3.5. ■

Definition of T and proof of Item 3. We also define the promised mapping $T(S)$ in the same manner as in the proof of Lemma 6.3.1, but using V' in place of V to decide whether or not we select an element x for inclusion in the set $F(S)$.

To establish Property 3, we note that the proofs of Claims 6.3.6 and 6.3.7 do not rely on the submodularity of f, therefore they apply as-is to the case where we compute on $\widetilde{f}$, even though $\widetilde{f}$ is not necessarily submodular.

Property 4 also follows as in the proof of Lemma 6.3.1. This completes the proof of the Lemma. ■

We now present our algorithm for learning monotone submodular functions over product distributions. For a subset of the universe $V \subseteq U$, let $\mathcal{P}_V$ denote the distribution $\mathcal{P}$ restricted to the variables in V. Note that if $\mathcal{P}$ is a product distribution, then $\mathcal{P}_V$ remains a product distribution and is easy to sample from.

Learn$(f, \alpha, \beta, \mathcal{P})$:

1. Let $\rho = \frac{\alpha^2}{6\log(2/\beta)}$.
2. Construct the collection of functions $\mathcal{G}$ as well as the mappings F, V, T given by Lemma 6.3.8 with parameter ρ.
3. Estimate the value $\mu_{g^B} = \mathbb{E}_{S\sim\mathcal{P}_{V(B)\setminus T(B)}}[g^B(S)]$ for each $g^B \in \mathcal{G}$.

Output the data structure h that consists of the values μ_{g^B} for every $g^B \in \mathcal{G}$ as well as the mapping F.

Figure 6.3: Learning a monotone submodular function

Theorem 6.3.12. *For any $\alpha, \beta \in (0,1]$, Algorithm 6.5 (α,β)-approximates any submodular function $f : 2^U \to [0,1]$ under any product distribution $\mathcal{P}$ in time $|U|^{O(\alpha^{-2}\log(1/\beta))}$ using oracle queries to f of tolerance $\alpha^2/72\log(2/\beta)$.*

Proof. For a set $S \subseteq U$, we let $B = F(S)$ and g^B be the corresponding submodular function as in Lemma 6.3.8. Note that since the queries have tolerance $\alpha^2/72\log(1/\beta) \leqslant \rho/12$, the lemma applies. We will analyze the error probability as if the estimates μ_{g^B} were computed using exact oracle queries to f, and will note that using tolerant queries to f can only introduce an additional

error of $\alpha^2/72\log(1/\beta) \leqslant \alpha/6$. We claim that, under this condition

$$\begin{aligned}\mathop{\mathbb{P}}_{S\sim\mathcal{P}}\{|f(S)-h(S)|>5\alpha/6\} &= \mathop{\mathbb{P}}_{S\sim\mathcal{P}}\left\{\left|g^{F(S)}(S)-\mu_{g^{F(S)}}\right|>5\alpha/6\right\}\\ &= \sum_{g^B\in\mathcal{G}} \mathop{\mathbb{P}}_{S\sim\mathcal{P}}\{B=F(S)\}\cdot \mathop{\mathbb{P}}_{S\sim\mathcal{P}}\left\{\left|g^B(S)-\mu_{g^B}\right|>5\alpha/6 \mid B=F(S)\right\}.\end{aligned} \tag{6.6}$$

To see this, recall that for every $S\subseteq U$, $g^{F(S)}(S)=f(S)$. By Property 3 of Lemma 6.3.1, the condition that $B=F(S)$ is equivalent to the conditions that $B\subseteq S\subseteq V(B)$ and $S\cap T(B)=\emptyset$. Hence,

$$\mathop{\mathbb{P}}_{S\sim\mathcal{P}}\left\{\left|g^B(S)-\mu_{g^B}\right|>5\alpha/6 \mid B=F(S)\right\} = \mathop{\mathbb{P}}_{S\sim\mathcal{P}_{V(B)\setminus T(B)}}\left\{\left|g^B(S)-\mu_{g^B}\right|>5\alpha/6\right\}.$$

Now, applying the concentration inequality for submodular functions stated as Corollary 6.2.6, we get

$$\mathop{\mathbb{P}}_{S\sim\mathcal{P}_{V_B\setminus T_B}}\left\{\left|g^B(S)-\mu_{g^B}\right|\geqslant\rho t\right\}\leqslant 2\exp\left(-\frac{t^2}{2(1/\rho+5/6t)}\right). \tag{6.7}$$

Plugging in $t=5\alpha/6\rho=\frac{5\log(2/\beta)}{\alpha}$ and simplifying we get $\mathbb{P}_{S\sim\mathcal{P}_{V(B)\setminus T(B)}}\left\{\left|g^B(S)-\mu_{g^B}\right|>\alpha\right\}\leqslant\beta$. Combining this with (6.6), the claim follows. ∎

6.3.2 Non-monotone submodular functions

For non-monotone functions, we need a more refined argument. Our main structure theorem replaces Property 1 in Lemma 6.3.1 by the stronger guarantee that $|\partial_x g(S)|\leqslant\alpha$ for all $g\in\mathcal{G}$, even for non-monotone submodular functions. Observe that for a submodular function $f:2^V\to\mathbb{R}$, the function $\overline{f}:2^V\to\mathbb{R}$ defined as $\overline{f}(S)=f(V\setminus S)$ is also submodular; moreover

$$\inf_{x\in V,S\subseteq V}\partial_x\overline{f}(S)=-\sup_{x\in V,S\subseteq V}\partial_x f(S). \tag{6.8}$$

Given these two facts, we can now prove our main structure theorem.

Theorem 6.3.13. *Given any submodular function $f:2^U\to[0,1]$ and $\rho>0$, Algorithm 6.4 makes the following guarantee. There are maps $F:2^U\to 2^U\times 2^U$ and $T:2^U\times 2^U\to 2^U$ such that:*

1. (Lipschitz) *For every $g^{B,C}\in\mathcal{G}$, $g^{B,C}$ is submodular and satisfies $\sup_{x\in V(B,C),S\subseteq V(B,C)}|\partial_x g^{B,C}(S)|\leqslant\rho$.*

Input: Tolerant oracle access to a submodular function $f : 2^U \to [0,1]$ with tolerance at most $\rho/12$ and parameter $\rho > 0$.

Let $\widetilde{f}$ denote the function specified by tolerant oracle queries to f such that for every $S \subseteq U$, $|f(S) - \widetilde{f}(S)| \leqslant \rho/12$.
Let $\prec$ denote an arbitrary ordering of U.
Let $\mathcal{G}(f)$ denote the collection of functions returned by Algorithm 6.2 with oracle f and parameter ρ, and let F_f, V_f, T_f be the associated mappings promised by Lemma 6.3.8.
for $g^B \in \mathcal{G}(f)$ **do**
  Let $\mathcal{G}(B)$ be the collection of functions returned by Algorithm 6.2 with oracle $\overline{g}^B$ and parameter ρ, and let F_B, V_B, T_B be the associated mappings given by Lemma 6.3.8.
end for
Let $V(S,T) = V_f(S) \cap V_S(T)$ denote the set of elements that have small marginal absolute value with respect to $S, T \subseteq U$.

Output: the collection of functions $\mathcal{G} = \bigcup_{g^B \in \mathcal{G}(f)} \{g^{B,C} = \overline{g}^C \mid g^C \in \mathcal{G}(B)\}$ where $g^{B,C} : 2^{V(B,C)} \to [0,1]$.

Figure 6.4: Decomposition for submodular functions from tolerant queries (Algorithm 6.4)

2. (Completeness) *For every* $S \subseteq U$, $F(S) \subseteq S \subseteq V(F(S))$ *and* $g^{F(S)}(S) = f(S)$.

3. (Uniqueness) *For every* $g^{B,C} \in \mathcal{G}$, $F(S) = (B,C)$ *if and only if* $B, C \subseteq S \subseteq V(B,C)$ *and* $S \cap T(B,C) = \emptyset$.

4. (Size) *The size of* $\mathcal{G}$ *is at most* $|\mathcal{G}| = |U|^{O(1/\rho)}$. *Moreover, given tolerant oracle access to* f *with tolerance* $\rho/12$, *we compute* F, V, T *in time* $|U|^{O(1/\rho)}$.

Proof. First we show Item 1.

Claim 6.3.14 (Lipschitz). *For every* $g^{B,C} \in \mathcal{G}$, $g^{B,C}$ *is submodular and*

$$\sup_{x \in V(B,C), S \subseteq V(B,C)} |\partial_x g^{B,C}(S)| \leqslant \rho.$$

Proof. Submodularity follows directly from Property 1 of Lemma 6.3.8. The same property of the lemma guarantees that for every $g^B \in \mathcal{G}(f)$ and $g^C \in \mathcal{G}(B)$, $\sup_{x \in V_B(C), S \subseteq V_B(C)} \partial_x g^C(S) \leqslant \rho$. Moreover, by (6.8), $\inf_{x \in V_f(B), S \subseteq V_f(B)} \partial_x \overline{g}^B(S) \geqslant -\rho$. Taken together, we obtain $\sup_{x \in V(B,C), S \subseteq V(B,C)} |\partial_x g^{B,C}(S)| \leqslant \rho$. ∎

Definition of F and proof of Item 2. Item 2 will follow from the analogous property in Lemma 6.3.8 almost directly. To construct the mapping $F(S)$, we want to first compute the appropriate function $g^B \in \mathcal{G}(f)$, using $F_f(S)$ and then find the appropriate function $g^C \in \mathcal{G}(B)$ using $F_B(S)$. Thus we can take $F(S) = (F_f(S), F_{F_f(S)}(S))$. By Lemma 6.3.8, Item 2 we have $B \subseteq S \subseteq V_f(B)$ and $C \subseteq S \subseteq V_B(C)$, so we conclude $B, C \subseteq S \subseteq V(B, C)$.

Definition of T and proof of Item 3. Item 3 will also follow from the analogous property in Lemma 6.3.8. By Lemma 6.3.8, Item 3, we have that $F_f(S) = B$ if and only if $B \subseteq S \subseteq V_f(B)$ and $S \cap T_f(B) = \emptyset$. By the same Lemma, we also have that $F_B(S) = C$ if and only if $C \subseteq S \subseteq V_B(C)$ and $S \cap T_B(C) = \emptyset$. So if we define $T(B, C) = T_f(B) \cup T_B(C)$, we can conclude that $F(S) = (B, C)$ if and only if $B, C \subseteq S \subseteq V(B, C)$ and $S \cap T(B, C) = \emptyset$.

Now it is clear that $F(S) = (B, C)$ if $F_f(S) = B$ and $F_B(S) = C$, which by Item 3 of Lemma 6.3.8 necessitates that $B \subseteq S \subseteq V_f(B)$, $S \cap T_f(B) = \emptyset$, $C \subseteq S \subseteq V_B(S)$, and $S \cap T_B(C) = \emptyset$. We have already defined $V(B, C)$ and now we define $T(B, C) = T_f(B) \cup T_B(C)$. It is clear now that $F(S) = (B, C)$ if and only if $B, C \subseteq S \subseteq V(B, C)$ and $S \cap T(B, C) = \emptyset$.

The size of $\mathcal{G}$ and running time bounds in Item 4 also follow directly from the analogous property of Lemma 6.3.1. The fact that we can compute the family $\mathcal{G}$ and the associated mappings F, V, T using oracle access to f with tolerance $\rho/12$ follows from the fact that each invocation of Lemma 6.3.1 can be computed using queries with tolerance $\rho/12$ and from the fact that Algorithm 6.4 only queries f in order to invoke Lemma 6.3.8. This completes the proof of the theorem. ■

We now present our algorithm for learning arbitrary submodular functions over product distributions. For a subset of the universe $V \subseteq C$, let $\mathcal{D}_V$ denote the distribution $\mathcal{D}$ restricted to the variables in V. Note that if $\mathcal{D}$ is a product distribution, then $\mathcal{D}_V$ remains a product distribution and is easy to sample from.

Learn$(f, \alpha, \beta, \mathcal{D})$:

Let $\rho = \frac{\alpha^2}{6\log(2/\beta)}$.

Construct the collection of functions $\mathcal{G}$ and the associated mappings F, V, T given by Lemma 6.3.1 with parameter ρ.

Estimate the value $\mu_{g^{B,C}} = \mathbb{E}_{S \sim \mathcal{D}_{V(B,C) \setminus T(B,C)}}[g^{B,C}(S)]$ for each $g^{B,C} \in \mathcal{G}$.

Output the data structure h that consists of the values $\mu_{g^{B,C}}$ for every $g^{B,C} \in \mathcal{G}$ as well as the mapping F.

Figure 6.5: Learning a non-monotone submodular function

To avoid notational clutter, throughout this section we will not consider the details of how we construct our estimate μ_g. However, it is an easy observation that this quantity can be estimated to a sufficiently high degree of accuracy using a small number of random samples.

Theorem 6.3.15. *For any $\alpha, \beta \in (0,1]$, Algorithm 6.5 (α,β)-approximates any submodular function $f\colon 2^U \to [0,1]$ under any product distribution in time $|U|^{O(\alpha^{-2}\log(1/\beta))}$ using oracle queries to f of tolerance $\alpha^2/72\log(1/\beta)$*

Proof. For a set $S \subseteq U$, we let $(B,C) = F(S)$ and $g^{B,C}$ be the corresponding submodular function as in Theorem 6.3.13. Note that since the queries have tolerance $\alpha^2/72\log(1/\beta) \leqslant \rho/12$, the lemma applies. We will analyze the error probability as if the estimates μ_{g^B} were computed using exact oracle queries to f, and will note that using tolerant queries to f can only introduce an additional error of $\alpha^2/72\log(1/\beta) \leqslant \alpha/6$. We claim that,

$$\begin{aligned}
&\mathop{\mathbb{P}}_{S\sim\mathcal{D}}\{|f(S)-h(S)| > 5\alpha/6\} \\
&= \mathop{\mathbb{P}}_{S\sim\mathcal{D}}\left\{\left|g^{F(S)}(S) - \mu_{g^{F(S)}}\right| > 5\alpha/6\right\} \\
&= \sum_{g^{B,C}\in\mathcal{G}} \mathop{\mathbb{P}}_{S\sim\mathcal{D}}\{(B,C) = F(S)\} \cdot \mathop{\mathbb{P}}_{S\sim\mathcal{D}}\left\{\left|g^{B,C}(S) - \mu_{g^{B,C}}\right| > 5\alpha/6 \mid (B,C) = F(S)\right\}. \quad (6.9)
\end{aligned}$$

To see this, recall that for every $S \subseteq U$, $g^{F(S)}(S) = f(S)$. By Property 3 of Lemma 6.3.1, the condition that $B = F(S)$ is equivalent to the conditions that $B, C \subseteq S \subseteq V(B,C)$ and $S \cap T(B,C) = \emptyset$. Hence,

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}}\left\{\left|g^{B,C}(S) - \mu_{g^{B,C}}\right| > 5\alpha/6 \mid (B,C) = F(S)\right\} = \mathop{\mathbb{P}}_{S\sim\mathcal{D}_{V(B,C)\setminus T(B,C)}}\left\{\left|g^{B}(S) - \mu_{g^{B}}\right| > 5\alpha/6\right\}.$$

Now, applying the concentration inequality for submodular functions stated as Corollary 6.2.6, we get

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}_{V(B,C)\setminus T(B,C)}}\left\{\left|g^{B,C}(S) - \mu_{g^{B,C}}\right| \geqslant \rho t\right\} \leqslant 2\exp\left(-\frac{t^2}{2(1/\rho + 5t/6)}\right). \quad (6.10)$$

Plugging in $t = 5\alpha/6\rho = \frac{5\log(2/\beta)}{\alpha}$ and simplifying we get $\mathbb{P}_{S\sim\mathcal{D}_{V(B,C)\setminus T(B,C)}}\left\{\left|g^{B,C}(S) - \mu_{g^{B,C}}\right| > \alpha\right\} \leqslant \beta$. Combining this with Equation (6.9), the claim follows. ∎

6.4 Applications to privacy-preserving query release

In this section, we show how to apply our algorithm from Section 6.3 to the problem of releasing monotone conjunctions over a boolean database. In

Section 6.4.1, we also show how our mechanism can be applied to release the *cut function* of an arbitrary graph.

Let us now begin with the monotone disjunctions. We will then extend the result to monotone conjunctions. Given our previous results, we only need to argue that monotone disjunctions can be described by a submodular function. Indeed, every element $S \in \{0,1\}^d$ naturally corresponds to a monotone Boolean disjunction $d_S : \{0,1\}^d \to \{0,1\}$ by putting

$$d_S(x) \stackrel{\text{def}}{=} \bigvee_{i\colon S(i)=1} x_i .$$

Note that in contrast to Section 6.3 here we use x to denote an element of $\{0,1\}^d$. Let $F_{\text{Disj}} : \{0,1\}^d \to [0,1]$ be the function such that $F_{\text{Disj}}(S) = d_S(D)$. It is easy to show that $F_{\text{Disj}}(S)$ is a monotone submodular function.

Lemma 6.4.1. *F_{Disj} is a monotone submodular function.*

Proof. Let X_i^+ denote the set of elements $x \in D$ such that $x_i = 1$, and let X_i^- denote the set of elements $x \in D$ such that $x_i = 0$. Consider the set system $U = \{X_i^+, X_i^-\}_{i=1}^d$ over the universe of elements $x \in D$. Then there is a natural bijection between $F_{\text{Disj}}(D)$ and the set coverage function $\text{Cov} : 2^U \to [0, |D|]$ defined to be $\text{Cov}(S) = |\bigcup_{X \in U} X|$, which is a monotone submodular function. ∎

We therefore obtain the following corollary directly by combining Theorem 6.3.15 with Proposition 2.9.1.

Corollary 6.4.2. *Let $\alpha, \beta, \varepsilon > 0$. There is an ε-differentially private algorithm that (α, β)-releases the set of monotone Boolean disjunctions over any product distribution in time $d^{t(\alpha,\beta)}$ for any data set of size $|D| \geqslant d^{t(\alpha,\beta)}/\varepsilon$ where $t(\alpha, \beta) = O(\alpha^{-2} \log(1/\beta))$.*

For completeness, we will present the algorithm for privately releasing monotone disjunctions over a product distribution $\mathcal{P}$ for a data set D, though we will rely on Corollary 6.4.2 for the formal analysis.

We will next see that this corollary directly transfers to monotone conjunctions. A monotone Boolean conjunction $c_S : \{0,1\}^d \to \{0,1\}$ is defined as

$$c_S(x) \stackrel{\text{def}}{=} \bigwedge_{i \in S} x_i = 1 - \bigvee_{i \in S} (1 - x_i).$$

Given the last equation, it is clear that in order to release conjunctions over some distribution, it is sufficient to release disjunctions over the same distribution after replacing every data item $x \in D$ by its negation $\bar{x}$, i.e., $\bar{x}_i = 1 - x_i$. Hence, Corollary 6.4.2 extends directly to monotone conjunctions.

Release$(D, \alpha, \beta, \varepsilon, \mathcal{P})$:

1. **Simulate** the oracle queries $F_{Disj}(S)$ by answering with $d_S(D) + Lap(1/\varepsilon|D|)$.
2. Let $\rho = \frac{\alpha^2}{6\log(2/\beta)}$.
3. **Construct** the collection of functions $\mathcal{G}$ and the associated mappings F, V, T given by Lemma 6.3.1 on the function F_{Disj} with parameter ρ.
4. **Estimate** the value $\mu_{g^B} = \mathbb{E}_{S\sim\mathcal{P}_{V(B)\setminus T(B)}}[g^B(S)]$ for each $g^B \in \mathcal{G}$.

Output the data structure h that consists of the estimated values μ_{g^B} for every $g^B \in \mathcal{G}$ as well as the mapping F. To evaluate any monotone disjunction query $d_S(D)$, compute $\mu_{g^{F(S)}}$.

Figure 6.6: Privately releasing monotone disjunctions

Extension to width w. Note that the uniform distribution on disjunctions of width w is not a product distribution, which is what we require to apply Theorem 6.3.15 directly. However, in Lemma 6.4.3 we show that for monotone submodular functions (such as F^D_{Disj}) the concentration of measure property required in the proof Theorem 6.3.15 is still satisfied. Of course, we can instantiate the theorem for every $w \in \{1, \ldots, k\}$ to obtain a statement for conjunctions of any width.

Indeed, given a monotone submodular function $f : 2^U \to \mathbb{R}$, let $S \in 2^U$ be the random variable where for every $x \in U$, independently $x \in S$ with probability w/d and $x \notin S$ with probability $1 - w/d$. On the other hand, let $T \in 2^U$ denote the uniform distribution over strings in 2^U of weight w. The following lemma is due to Balcan and Harvey [BH].

Lemma 6.4.3. *Assume $f : 2^U \to \mathbb{R}$ is monotone function, and S and T are chosen at random as above. Then,*

$$\mathbb{P}[f(T) \geqslant \tau] \leqslant 2\,\mathbb{P}[f(S) \geqslant \tau] \tag{6.11}$$

$$\mathbb{P}[f(T) \leqslant \tau] \leqslant 2\,\mathbb{P}[f(S) \leqslant \tau] \tag{6.12}$$

Remark 6.4.4. Throughout this section we focus on the case of *monotone* disjunctions and conjunctions. Our algorithm can be extended to non-monotone conjunctions/disjunctions as well. However, this turns out to be less interesting than the monotone case. Indeed, a random non-monotone conjunction of width w is false on any fixed data item with probability 2^{-w}, thus when

$w \geqslant \log(1/\alpha)$, the constant function 0 is a good approximation to F_{Disj} on a random non-monotone conjunction of width w. We therefore omit the non-monotone case from our presentation.

6.4.1 Releasing the cut function of a graph

Consider a graph $G = (V, E)$ in which the edge-set represents the private database (We assume here that each individual is associated with a single edge in G. The following discussion generalizes to the case in which individuals may be associated with multiple edges, with a corresponding increase in sensitivity). The *cut function* associated with G is $f_G : 2^V \to [0,1]$, defined as:

$$f_G(S) = \frac{1}{|V|^2} \cdot |\{(u,v) \in E : u \in S, v \notin S\}|$$

We observe that the graph cut function encodes a collection of counting queries over the database E and so has sensitivity $1/|V|^2$.

Fact 6.4.5. *For any graph G, f_G is submodular.*

Lemma 6.4.6. *The decomposition from Theorem 6.3.13 constructs a collection of functions $\mathcal{G}$ of size $|\mathcal{G}| \leqslant 2^{2/\alpha}$.*

Proof. Let $u \in V$, and $S \subset V$ such that $|\partial_u f_G(S)| \geqslant \alpha$. It must be that the degree of u in G is at least $\alpha \cdot |E|$. But there can be at most $2/\alpha$ such high-influence vertices, and therefore at most $2^{2/\alpha}$ subsets of high influence vertices. ∎

Corollary 6.4.7. *Algorithm 6.5 can be used to privately (α, β)-release the cut function on any graph over any product distribution in time $t(\alpha, \beta, \varepsilon)$ for any database of size $|D| \geqslant t(\alpha, \beta, \varepsilon)$, while preserving ε-differential privacy, where:*

$$t(\alpha, \beta, \varepsilon) = \frac{2^{O(\alpha^{-2}\log(1/\beta))}}{\varepsilon}$$

Proof. This follows directly from a simple modification of Theorem 6.3.15, by applying Lemma 6.4.6 and plugging in the size of the decomposition $\mathcal{G}$. The algorithm can then be made privacy preserving by applying proposition 2.9.1. ∎

6.5 Equivalence between agnostic learning and query release

In this section we show an information-theoretic equivalence between *agnostic learning* and *query release* in the statistical queries model. In particular, given

an agnostic learning algorithm for a specific concept class we construct a query release algorithm for the same concept class.

Consider a distribution A over $X \times \{0,1\}$ and a concept class C. An *agnostic learning* algorithm (in the strong sense) finds the concept $q \in C$ that approximately maximizes $\mathbb{P}_{(x,b)\sim A}\{q(x) = b\}$ to within an additive error of α. Our reduction from query release to agnostic learning actually holds even for *weak agnostic learning*. A weak agnostic learner is not required to maximize $\mathbb{P}_{(x,b)\sim A}\{q(x) = b\}$, but only to find a sufficiently good predicate q provided that one exists.

Definition 6.5.1 (Weak Agnostic SQ-Learning)**.** Let C be a concept class and $\gamma, \tau > 0$ and $0 < \beta < \alpha \leqslant 1/2$. An algorithm $\mathcal{A}$ with oracle access to $\mathrm{STAT}_\tau(A)$ is an $(\alpha, \beta, \gamma, \tau)$-*weak agnostic learner for* C if for every distribution A such that there exists $q^* \in C$ satisfying $\mathbb{P}_{(x,b)\sim A}\{q^*(x) = b\} \geqslant 1/2 + \alpha$, $\mathcal{A}(A)$ outputs a predicate $q : X \to \{0,1\}$ such that $\mathbb{P}_{(x,b)\sim A}\{q(x) = b\} \geqslant 1/2 + \beta$, with probability at least $1 - \gamma$.

Note that if we can agnostically learn C in the strong sense from queries of tolerance τ to within additive error $\alpha - \beta$ with probability $1 - \gamma$, then there is also an $(\alpha, \beta, \gamma, \tau)$-weak agnostic learner.

We are now ready to state the main result of this section, which shows that a weak agnostic SQ-learner for any concept class is sufficient to release the same concept class in the SQ model.

Theorem 6.5.2. *Let C be a concept class. Let $\mathcal{A}$ be an algorithm that $(\alpha/2, \beta, \gamma, \tau)$ weak agnostic-SQ learns C with $\tau \leqslant \beta/8$. Then there exists an algorithm $\mathcal{B}$ that invokes $\mathcal{A}$ at most $T = 8\log|X|/\beta^2$ times and $(\alpha, 0)$-releases C with probability at least $1 - T\gamma$.*

The proof strategy is as follows. We will start from D_0 being the uniform distribution over X. We will then construct a short sequence of distributions $D_1, D_2, \ldots, D_T$ such that no concept in C can distinguish between D and D_T up to bias α. Each distribution D_t is obtained from the previous one using a multiplicative weights approach as in Chapter 4 and with the help of the learning algorithm that's given in the assumption of the theorem. Intuitively, at every step we use the agnostic learner to give us the predicate $q_t \in C$ which distinguishes the most between D_t and D. In order to accomplish this we feed the agnostic learner with the distribution A_t that labels elements sampled from D by 1 and elements sampled from D_t by 0. For a technical reason we also need to consider the distribution with 0 and 1 flipped. Once we obtained q_t we can use it as a penalty function in the update rule of the multiplicative weights method. This has the effect of bringing D and D_t closer in relative entropy. A typical potential argument then bounds the number of update

Let D_0 denote the uniform distribution over X.
For $t = 1, \ldots, T = \lceil 8 \log |X| / \beta^2 \rceil + 1$:

1. Consider the distributions

$$A_t^+ = 1/2(D, 1) + 1/2(D_{t-1}, 0) \qquad A_t^- = 1/2(D, 0) + 1/2(D_{t-1}, 1).$$

   Let $q_t^+ = \mathcal{A}(A_t^+)$ and $q_t^- = \mathcal{A}(A_t^-)$. Let v_t^+ be the value returned by $\mathrm{STAT}_\tau(A_t^+)$ on the query q_t^+ and v_t^- be the value returned by $\mathrm{STAT}_\tau(A_t^-)$ on the query q_t^-. Let $v_t = \max\{v_t^+, v_t^-\} - 1/2$ and q_t be the corresponding query.

2. If

$$v_t \leqslant \frac{\beta}{2} - \tau, \tag{6.13}$$

   then proceed to "output" step.

3. **Update:** Let D_t be the distribution obtained from D_{t-1} using a multiplicative weights update step with penalty function induced by q_t and penalty parameter $\eta = \beta/2$ as follows:

$$D_t'(x) = \exp(\eta q_t(x)) \cdot D_{t-1}(x)$$

$$D_t(x) = \frac{D_t'}{\sum_{x \in X} D_t'(x)}$$

Output: $a_c = \mathbb{E}_{x \sim D_T} c(x)$ for each $c \in C$.

Figure 6.7: Data release via agnostic learning

steps that can occur before we reach a distribution D_t for which no good distinguisher in C exists.

6.5.1 Proof of Theorem 6.5.2

Proof. We start by relating the probability that q_t predicts b from x on the distribution A_t^+ to the difference in expectation of q_t on D and D_{t-1}.

Lemma 6.5.3. *For any* $q \colon X \to \{0, 1\}$,

$$\mathop{\mathbb{P}}_{(x,b) \sim A_t^+} \{q(x) = b\} - \frac{1}{2} = \frac{1}{2} \left(\mathop{\mathbb{E}}_{x \sim D} q(x) - \mathop{\mathbb{E}}_{x \sim D_{t-1}} q(x) \right) \tag{6.14}$$

Proof. If $q_t = q_t^+$ then

$$\begin{aligned}\mathop{\mathbb{P}}_{(x,b)\sim A_t^+}\{q(x)=b\} &= \frac{1}{2}\mathop{\mathbb{P}}_{x\sim D}\{q(x)=1\}+\frac{1}{2}\mathop{\mathbb{P}}_{x\sim D_{t-1}}\{q(x)=0\}\\ &= \frac{1}{2}\mathop{\mathbb{E}}_{x\sim D}[q(x)]+\frac{1}{2}\mathop{\mathbb{E}}_{x\sim D_{t-1}}[1-q(x)]\\ &= \frac{1}{2}+\frac{1}{2}\left(\mathop{\mathbb{E}}_{x\sim D}q(x)-\mathop{\mathbb{E}}_{x\sim D_{t-1}}q(x)\right)\end{aligned}$$

Note that $\mathbb{P}_{(x,b)\sim A_t^-}\{q(x)=b\} = 1-\mathbb{P}_{(x,b)\sim A_t^-}\{q(x)=(1-b)\} = 1-\mathbb{P}_{(x,b)\sim A_t^+}\{q(x)=b\}$, so if $q_t = q_t^-$ then

$$\begin{aligned}\mathop{\mathbb{P}}_{(x,b)\sim A_t^+}\{q(x)=b\} = 1-\mathop{\mathbb{P}}_{(x,b)\sim A_t^-}\{q(x)=b\} &= 1-\left(\frac{1}{2}-\frac{1}{2}\left(\mathop{\mathbb{E}}_{x\sim D}q(x)-\mathop{\mathbb{E}}_{x\sim D_{t-1}}q(x)\right)\right)\\ &= \frac{1}{2}+\frac{1}{2}\left(\mathop{\mathbb{E}}_{x\sim D}q(x)-\mathop{\mathbb{E}}_{x\sim D_{t-1}}q(x)\right)\end{aligned}$$

■

The proof closely follows the utility analysis in Chapter 4. For two distributions P, Q on a universe X we define the *relative entropy* to be $\mathrm{RE}(P\|Q) = \sum_{x\in X} P(x)\log(P(x)/Q(x))$. We consider the potential

$$\Psi_t = \mathrm{RE}(D\|D_t).$$

Fact 6.5.4. $\Psi_t \geqslant 0$

Fact 6.5.5. $\Psi_0 \leqslant \log|X|$

We will argue that in every step the potential drops by at least $\beta^2/4$. Hence, we know that there can be at most $4\log|X|/\alpha^2$ steps before we reach a distribution that satisfies (6.13).

The next lemma gives a lower bound on the potential drop in terms of the concept, q_t, returned by the learning algorithm at time t. Recall, that η (used below) is the penalty parameter used in the multiplicative weights update rule.

Lemma 6.5.6 (cf. Lemma 4.3.5)**.**

$$\Psi_{t-1}-\Psi_t \geqslant \eta\left|\mathop{\mathbb{E}}_{x\sim D}q_t(x)-\mathop{\mathbb{E}}_{x\sim D_{t-1}}q_t(x)\right|-\eta^2 \tag{6.15}$$

Let

$$\mathrm{opt}_t = \sup_{q\in C}\left|\mathop{\mathbb{P}}_{(x,b)\sim A_t^+}\{q(x)=b\}-\frac{1}{2}\right|.$$

Note that $\mathbb{P}_{(x,b)\sim A_t^-}\{q(x) = b\} = 1 - \mathbb{P}_{(x,b)\sim A_t^+}\{\neg q(x) = b\}$. For the remainder of the proof we treat the two cases symmetrically and only look at how far from 1/2 these probabilities are. The next lemma shows that either opt_t is large or else we are done in the sense that D_t is indistinguishable from D for any concept from C.

Lemma 6.5.7. *Let $\alpha > 0$. Suppose*

$$\mathrm{opt}_t \leqslant \frac{\alpha}{2}.$$

Then, for all $q \in C$,

$$\left| \mathop{\mathbb{E}}_{x\sim D} q(x) - \mathop{\mathbb{E}}_{x\sim D_t} q_t(x) \right| \leqslant \alpha \tag{6.16}$$

Proof. From Lemma 6.5.3 we have that for every $q \in C$

$$\frac{\alpha}{2} \geqslant \mathrm{opt}_t \geqslant \mathop{\mathbb{P}}_{(x,b)\sim A_t^+}\{q(x) = b\} - \frac{1}{2} = \frac{1}{2}\left(\mathop{\mathbb{E}}_{x\sim D} q(x) - \mathop{\mathbb{E}}_{x\sim D_t} q_t(x) \right)$$

Thus $\alpha \geqslant \left(\mathbb{E}_{x\sim D}\, q(x) - \mathbb{E}_{x\sim D_t}\, q_t(x) \right)$. Similarly,

$$\frac{\alpha}{2} \geqslant \mathrm{opt}_t \geqslant \mathop{\mathbb{P}}_{(x,b)\sim A_t^-}\{q(x) = b\} - \frac{1}{2} = \frac{1}{2}\left(\mathop{\mathbb{E}}_{x\sim D_t} q(x) - \mathop{\mathbb{E}}_{x\sim D} q_t(x) \right)$$

Thus $-\alpha \leqslant \left(\mathbb{E}_{x\sim D}\, q(x) - \mathbb{E}_{x\sim D_t}\, q_t(x) \right)$. So we conclude $\alpha \geqslant \left| \mathbb{E}_{x\sim D}\, q(x) - \mathbb{E}_{x\sim D_t}\, q_t(x) \right|$. ∎

We can now finish the proof of Theorem 6.5.2. By our assumption, we have that so long as $\mathrm{opt}_t \geqslant \alpha/2$ the algorithm $\mathcal{A}$ produces a concept q_t such that with probability $1 - \gamma$

$$\left| \mathop{\mathbb{P}}_{(x,b)\sim A_t^+}\{q_t(x) = b\} - \frac{1}{2} \right| \geqslant \beta. \tag{6.17}$$

For the remainder of the proof we assume that our algorithm returns a concept satisfying Equation 6.17 in every stage for which $\mathrm{opt}_t \geqslant \alpha/2$. By a union bound over the stages of the algorithm, this event occurs with probability at least $1 - T\gamma$.

Assuming Equation 6.13 is not satisfied we have that

$$\frac{\beta}{4} \leqslant \frac{\beta}{2} - 2\tau \leqslant v_t - \tau \leqslant \left| \mathop{\mathbb{P}}_{A_t^+}\{q_t(x) = b\} \right|.$$

The leftmost inequality follows because $\tau \leqslant \beta/8$. We then get

$$\begin{aligned}
\Psi_{t-1} - \Psi_t &\geqslant \eta \left| \mathop{\mathbb{E}}_{D} q_t(x) - \mathop{\mathbb{E}}_{D_{t-1}} q_t(x) \right| - \eta^2 && \text{(Lemma 6.5.6)} \\
&\geqslant \eta \left| 4 \mathop{\mathbb{P}}_{A_t}\{q_t(x) = b\} - 2 \right| - \eta^2 && \text{(Lemma 6.5.3)} \\
&\geqslant \eta \cdot \beta - \eta^2 && \text{(Equation 6.13 not satisfied)} \\
&\geqslant \frac{\beta^2}{2} - \frac{\beta^2}{4} && (\eta = \beta/2) \\
&= \frac{\beta^2}{4}
\end{aligned}$$

Hence, if we put $T \geqslant 4 \log |X|/\beta^2$, we must reach a distribution that satisfies (6.13). But at that point, call it t, the subroutine $\mathcal{A}$ outputs a concept q_t such that

$$\left| \mathop{\mathbb{P}}_{(x,b)\sim A_t^+} (q_t(x) = b) - \frac{1}{2} \right| \leqslant v_t + \tau < \frac{\beta}{2} + \tau < \beta$$

In this case, by our assumption that Equation 6.17 is satisfied whenever $\mathrm{opt}_t \geqslant 1/2 + \alpha/2$, we conclude that $\mathrm{opt}_t < 1/2 + \alpha/2$. By Lemma 8.2.4, we get

$$\sup_{q \in C} \left| \mathop{\mathbb{E}}_{x\sim D} q(x) - \mathop{\mathbb{E}}_{x\sim D_t} q_t(x) \right| \leqslant \alpha\,.$$

But this is what we wanted to show, since it means that our output on all concepts in C will be accurate up to error α. ∎

We remark that for clarity, we let the failure probability of the release algorithm grow linearly in the number of calls we made to the learning algorithm (by the union bound). However, this is not necessary: we could have driven down the probability of error in each stage by independent repetition of the agnostic learner.

This equivalence between release and agnostic learning also can easily be seen to hold in the reverse direction as well.

Theorem 6.5.8. *Let C be a concept class. If there exists an algorithm $\mathcal{B}$ that $(\alpha, 0)$-releases C with probability $1 - \gamma$ and accesses the database using at most k oracle accesses to $STAT_\tau(A)$, then there is an algorithm that makes $2k$ queries to $\mathrm{STAT}_\tau(A)$ and agnostically learns C in the strong sense with accuracy 2α with probability at least $1 - 2\gamma$.*

Proof. Let Y denote the set of examples with label 1, and let N denote the set of examples with label 0. We use $\mathrm{STAT}_\tau(A)$ to simulate oracles $\mathrm{STAT}_\tau(Y)$ and

$\mathrm{STAT}_\tau(N)$ that condition the queried concept on the label. That is, $\mathrm{STAT}_\tau(Y)$, when invoked on concept q, returns an approximation to $\mathbb{P}_{x\sim A}\{q(x) = 1 \wedge (x \in Y)\}$ and $\mathrm{STAT}_\tau(N)$ returns an approximation to $\mathbb{P}_{x\sim A}\{q(x) = 1 \wedge (x \in Y)]$. We can simulate a query to either oracle using only one query to $\mathrm{STAT}_\tau(A)$.

Run $\mathcal{B}(Y)$ to obtain answers $a_1^Y, \ldots, a_{|C|}^Y$, and run $\mathcal{B}(N)$ to obtain answers $a_1^N, \ldots, a_{|C|}^N$. Note that this takes at most $2k$ oracle queries, using the simulation described above, by our assumption on $\mathcal{B}$. By the union bound, except with probability 2γ, we have for all $q_i \in C$: $|q_i(Y) - a_i^Y| \leqslant \alpha$ and $|q_i(B) - a_i^N| \leqslant \alpha$. Let $q^* = \arg\max_{q_i \in C}(a_i^Y - a_i^N)$. Observe that $q^*(D) \geqslant \max_{q\in C} q(D) - 2\alpha$, and so we have agnostically learned C up to error 2α. ■

Feldman proves that even monotone conjunctions cannot be agnostically learned to subconstant error with polynomially many SQ queries:

Theorem 6.5.9 ([Fel]). *Let C be the class of monotone conjunctions. Let $k(d)$ be any polynomial in d, the dimension of the data space. There is no algorithm $\mathcal{A}$ which agnostically learns C to error $o(1)$ using $k(d)$ queries to $STAT_{1/k(d)}$.*

Corollary 6.5.10. *For any polynomial in d $k(d)$, no algorithm that makes $k(d)$ statistical queries to a database of size $k(d)$ can release the class of monotone conjunctions to error $o(1)$.*

Note that formally, Corollary 6.5.10 only precludes algorithms which release the approximately correct answers to *every* monotone conjunction, whereas our algorithm is allowed to make arbitrary errors on a small fraction of conjunctions.

Remark 6.5.11. It can be shown that the lower bound from Corollary 6.5.10 in fact does *not* hold when the accuracy requirement is relaxed so that the algorithm may err arbitrarily on 1% of all the conjunctions. Indeed, there is an inefficient algorithm (runtime $\mathrm{poly}(2^d)$) that makes $\mathrm{poly}(d)$ statistical queries and releases random conjunctions up to a small additive error. The algorithm roughly proceeds by running multiplicative weights privately while sampling, say, 1000 random conjunctions at every step and checking if any of them have large error. If so, an update occurs. We omit the formal description and analysis of the algorithm.

We also remark that the proofs of Theorems 6.5.2 and 6.5.8 are not particular to the statistical queries model: we showed generically that it is possible to solve the query release problem using a small number of black-box calls to a learning algorithm, *without accessing the database except through the learning algorithm*. This has interesting implications for any class of algorithms that may make only restricted access to the database. For example, this also proves that if it is possible to agnostically learn some concept class C while

preserving ε-differential privacy (even using algorithms that do not fit into the SQ model), then it is possible to release the same class while preserving $T\varepsilon \approx \log|X|\varepsilon$-differential privacy.

Chapter 7

Private Data Release via Learning Thresholds

7.1 Introduction

In Chapters 3–5, we saw algorithms and techniques for differentially private data analysis that allow accurate statistics for a rich class of queries. The most important shortcoming of these algorithms is that the running time depends linearly on the size of the data universe. This is tolerable for some practical applications as we demonstrated in Chapter 5. However, when the data dimensionality is large, the universe size becomes prohibitive.

Thus, a central question in differentially private data analysis is to develop general techniques and algorithms that are *efficient* in the sense that the running time is polynomial (or at least sub-exponential) in the data dimensionality. While some computational hardness results are known [DNR+, UV, GHRU], they apply only to restricted classes of data release algorithms.

In Chapter 6 we made a first step towards this goal by presenting an algorithm for releasing Boolean conjunctions with small error on most conjunctions. Our algorithm was based on a reduction to learning submodular functions in a certain sense. In this chapter we will strengthen the connection between private data release and learning theory significantly. As a result we will obtain several new release mechanism that all share the characteristic that the running time is subexponential or even polynomial in the data dimensionality.

This Work. Our primary contribution is a computationally efficient new tool for privacy-preserving data release: a general reduction to the task of *learning thresholds of sums of predicates*. The class of predicates (for learning) in our reduction is derived directly from the class of queries (for data release).

At a high level, we draw a connection between data release and learning as follows. In the data release setting, one can view the *database as a function*: it maps queries in $\mathcal{Q}$ to answers in $[0,1]$. The data release goal is approximating this function on queries/examples in $\mathcal{Q}$. The challenge is doing so with only

bounded access to the database/function; in particular, we only allow access that preserves differential privacy. For example, this often means that we only get a bounded number of oracle queries to the database function with noisy answers.

At this high level there is a striking similarity to learning theory, where a standard goal is to efficiently learn/approximate a function given limited access to it, e.g. a bounded number of labeled examples or oracle queries. Thus a natural approach to data release is *learning the database function* using a computational learning algorithm.

While the approach is intuitively appealing at this high level, it faces immediate obstacles because of apparent incompatibilities between the requirements of learning algorithms and the type of "limited" access to data that are imposed by private data release. For example, in the data release setting a standard technique for ensuring differential privacy is adding noise, but many efficient learning algorithms fail badly when run on noisy data. As another example, for private data release, the number of (noisy) database accesses is often very restricted: e.g sub-linear, or at most quadratic in the database size. In the learning setting, on the other hand, it is almost always the case that the number of examples or oracle queries required to learn a function is *lower bounded* by its description length (and is often a large polynomial in the description length).

Our work explores the connection between learning and private data release. We

(i) give an efficient reduction that shows that, in fact, a general class of data release tasks *can* be reduced to related and natural computational learning tasks; and

(ii) instantiate this general reduction using new and known learning algorithms to obtain new computationally efficient differentially private data release algorithms.

Before giving more details on our reduction in Section 7.1.1, we briefly discuss its context and some of the ways that we apply/instantiate it. While the search for efficient differentially private data release algorithms is relatively new, there are decades of work in learning theory aimed at developing techniques and algorithms for computationally efficient learning, going back to the early work of Valiant [Val]. Given the high-level similarity between the two fields, leveraging the existing body of work and insights from learning theory for data release is a promising direction for future research; we view our reduction as a step in this direction. We note that our work is by no means the first to draw a connection between privacy-preserving data release and learning theory; as discussed in the "Related Work" section below, several

prior works used learning techniques in the data release setting. A novelty in our work is that it gives an explicit and modular reduction from data release to natural learning problems. Conceptually, our reduction overcomes two main hurdles:

— bridging the gap between the *noisy* oracle access arising in private data release and the *noise-free* oracle access required by many learning algorithms (including the ones we use).

— avoiding any dependence on the database size in the complexity of the learning algorithm being used.

We use this reduction to construct new data release algorithms. In this work we explore two main applications of our reduction. The first aims to answer boolean conjunction queries (also known as contingency tables or marginal queries), one of the most well-motivated and widely-studied classes of statistical queries in the differential privacy literature. Taking the data universe $\mathscr{U}$ to be $\{0,1\}^d$, the w-way boolean conjunction corresponding to a subset S of k attributes in $[d]$ counts what fraction of items in the database have all the attributes in S set to 1. Approximating the answers for w-way conjunctions (or all conjunctions) has been the focus of several past works (see, e.g. [BCD+, KRSU, UV, GHRU]). Applying our reduction with a new learning algorithm tailored for this class, we obtain a data release algorithm that, for databases of size $d^{O\left(\sqrt{w \log(w \log d)}\right)}$, releases accurate answers to all w-way conjunctions simultaneously (we ignore for now the dependence of the database size on other parameters such as the error). The running time is poly(d^w). Previous algorithms either had running time $2^{\Omega(d)}$ (e.g. [DNR+]) or required a database of size $d^{w/2}$ (adding independent noise [DMNS]). We also obtain better bounds for the task of approximating the answers to a large fraction of *all* (i.e. d-way) conjunctions under arbitrary distributions. These results follow from algorithms for learning thresholds of sums of the relevant predicates; we base these algorithms on learning theory techniques for representing such functions as low-degree polynomial threshold functions, following works such as [KS, KOS]. We give an overview of these results in Section 7.1.2 below.

Our second application uses Fourier analysis of the database (viewed, again, as a real-valued function on the queries in $\mathcal{Q}$). We obtain new polynomial and quasi-polynomial data release algorithms for parity counting queries and low-depth (AC^0) counting queries respectively. The learning algorithms we use for this are (respectively) Jackson's Harmonic Sieve algorithm [Jac], and an algorithm for learning Majority-of-AC^0 circuits due to Jackson *et al.* [JKS]. We elaborate on these results in Section 7.1.3 below.

7.1.1 Private data release reduces to learning thresholds

In this section we give more details on the reduction from privacy-preserving data release to learning thresholds. The full details are in Sections 7.2 and 7.3. We begin with loose definitions of the data release and learning tasks we consider, and then proceed with (a simple case of) our reduction.

Counting Queries, Data Release and Learning Thresholds. We refer to an element u in data domain $\mathscr{U}$ as an *item*. A *database* is a collection of n items from $\mathscr{U}$. A *counting query* is specified by a predicate $p : \mathscr{U} \to \{0,1\}$, and the query q_p on database D outputs the fraction of items in D that satisfy p, i.e. $\frac{1}{n}\sum_{i=1}^{n} p(D_i)$. A *class* of counting queries is specified by a set $\mathcal{Q}$ of query descriptions and a predicate $P\colon \mathcal{Q} \times \mathscr{U} \to \{0,1\}$. For a query $q \in \mathcal{Q}$, its corresponding predicate is $P(q,\cdot) : \mathscr{U} \to \{0,1\}$. We will sometimes fix a data item $u \in \mathscr{U}$ and consider the predicate $p_u(\cdot) \triangleq P(\cdot,u) : \mathcal{Q} \to \{0,1\}$.

Fix a data domain $\mathscr{U}$ and query class $\mathcal{Q}$ (specified by a predicate P). A *data release algorithm* $\mathscr{A}$ gets as input a database D, and outputs a *synopsis* $S : \mathcal{Q} \to [0,1]$ that provides approximate answers to queries in $\mathcal{Q}$. We say that $\mathscr{A}$ is an (α,β,γ) *distribution-free data release algorithm for* $(\mathscr{U},\mathcal{Q},P)$ if, for any distribution G over the query set $\mathcal{Q}$, with probability $1-\beta$ over the algorithm's coins, the synopsis S satisfies that with probability $1-\gamma$ over $q \sim G$, the (additive) error of S on q is bounded by α. Later we will also consider data release algorithms that only work for a specific distribution or class of distributions (in this case we will not call the algorithm distribution-free). Finally, we assume for now that the data release algorithm only accesses the distribution G by sampling queries from it, but later we will also consider more general types of access (see below). A *differentially private* data release algorithm is one whose output distribution (on synopses) is differentially private as per Definition 2.1.2. See Definition 7.2.3 for full and formal details.

Fix a class $\mathcal{Q}$ of *examples* and a set $\mathcal{F}$ of predicates on $\mathcal{Q}$. Let $\mathcal{F}_{n,t}$ be the set of thresholded sums from $\mathcal{F}$, i.e., the set of functions of the form $f = \mathbb{I}\left\{\frac{1}{n}\sum_{i=1}^{n} f_i \geqslant t\right\}$, where $f_i \in \mathcal{F}$ for all $1 \leqslant i \leqslant n$. We refer to functions in $\mathcal{F}_{n,t}$ as *n-thresholds*. An algorithm for learning thresholds gets access to a function in $\mathcal{F}_{n,t}$ and outputs a *hypothesis* $h : \mathcal{Q} \to \{0,1\}$ that labels examples in $\mathcal{Q}$. We say that it is a (γ,β) distribution-free learning algorithm for learning thresholds over $(\mathcal{Q},\mathcal{F})$ if, for any distribution G over the set $\mathcal{Q}$, with probability $1-\beta$ over the algorithm's coins the output hypothesis h satisfies that with probability $1-\gamma$ over $q \sim G$, h labels q correctly. As above, later we will also consider learning algorithms that are not distribution free, and only work for a specific distribution or class or distributions. For now, we assume that the learning algorithm only accesses the distribution G by drawing examples from it. These examples are labeled using the target function that the algorithm is trying to

learn. See Definition 7.2.4 for full and formal details.

The Reduction. We can now describe (a simple case of) our reduction from differentially private data release to learning thresholds. For any data domain $\mathcal{U}$, set $\mathcal{Q}$ of query descriptions, and predicate $P : \mathcal{Q} \times \mathcal{U} \to \{0,1\}$, the reduction shows how to construct a (distribution free) data release algorithm given a (distribution free) algorithm for learning thresholds over $(\mathcal{Q}, \{p_u : u \in \mathcal{U}\})$, i.e., any algorithm for learning thresholds where $\mathcal{Q}$ is the example set and the set $\mathcal{F}$ of predicates (over $\mathcal{Q}$) is obtained by the possible ways of fixing the u-input to P. The resulting data release algorithm is (α, β, γ)-accurate as long as the database is not too small; the size bound depends on the desired accuracy parameters and on the learning algorithm's sample complexity. The efficiency of the learning algorithm is preserved (up to mild polynomial factors).

Theorem 7.1.1 (Reduction from Data Release to Learning Thresholds, Simplified). *Let $\mathcal{U}$ be a data universe, $\mathcal{Q}$ a set of query descriptions, and $P : \mathcal{Q} \times \mathcal{U} \to \{0,1\}$ a predicate. There is an ε-differentially private (α, β, γ)-accurate distribution free data-release algorithm for $(\mathcal{U}, \mathcal{Q}, P)$, provided that:*

1. *there is a distribution-free learning algorithm $\mathscr{L}$ that (γ, β)-learns thresholds over $(\mathcal{Q}, \{p_u : u \in \mathcal{U}\})$ using $b(n, \gamma, \beta)$ labeled examples and running time $t(n, \gamma, \beta)$ for learning n-thresholds.*
2. $n \geqslant \frac{C \cdot b(n', \gamma', \beta') \cdot \log(1/\beta)}{\varepsilon \cdot \alpha \cdot \gamma}$, *where* $n' = \Theta(\log|\mathcal{Q}|/\alpha^2)$, $\beta' = \Theta(\beta \cdot \alpha)$, $\gamma' = \Theta(\gamma \cdot \alpha)$, $C = \Theta(1)$.

Moreover, the data release algorithm only accesses the query distribution by sampling. The number of samples taken is $O(b(n', \gamma', \beta') \cdot \log(1/\beta)/\gamma)$ and the running time is

$$\mathrm{poly}(t(n', \gamma', \beta'), n, 1/\alpha, \log(1/\beta), 1/\gamma).$$

Section 7.2.2 gives a formal (and more general) statement in Theorem 7.2.8. Section 7.2.3 gives a proof overview, and Section 7.3 gives the full proof. Note that, since the data release algorithm we obtain from this reduction is distribution free (i.e. works for any distribution on the query set) and only accesses the query distribution by sampling, it can be *boosted* to yield accurate answers on *all* the queries [DRV].

A More General Reduction. For clarity of exposition, we gave above a simplified form of the reduction. This assumed that the learning algorithm is *distribution-free* (i.e. works for any distribution over examples) and only requires sampling access to labeled examples. These strong assumptions enable

us to get a distribution-free data release algorithm that only accesses the query distribution by sampling.

We also give a reduction that applies even to distribution-specific learning algorithms that require (a certain kind of) oracle access to the function being learned. In addition to sampling labeled examples, the learning algorithm can: (i) estimate the distribution G on any example q by querying q and receiving a (multiplicative) approximation to the probability $G[q]$; and (ii) query an oracle for the function f being learned on any q such that $G[q] \neq 0$. We refer to this as *approximate distribution restricted oracle access*, see Definition 7.2.5. Note that several natural learning algorithms in the literature use oracle queries in this way; in particular, we show that this is true for Jackson's Harmonic Sieve Algorithm [Jac], see Section 7.5.

Our general reduction gives a data release algorithm for a class $\mathcal{GQ}$ of distributions on the query set, provided we have a learning algorithm which can also use approximate distribution restricted oracle access, and which works for a slightly richer class of distributions $\mathcal{GQ}'$ (a *smooth extension*, see Definition 7.2.7). Again, several such algorithms (based on Fourier analysis) are known in the literature; our general reduction allows us to use them and obtain the new data release results outlined in Section 7.1.3.

Related Work: Privacy and Learning. Our new reduction adds to the fruitful and growing interaction between the fields of differentially private data release and learning theory. Prior works also explored this connection. In our work, we "import" learning theory techniques by drawing a correspondence between the database (in the data release setting), for which we want to approximate query answers, and the target function (in the learning setting) which labels examples. Several other works have used this correspondence (implicitly or explicitly), e.g. [DNR+, DRV, GHRU]. A different view, in which *queries* in the data release setting correspond to *concepts* in learning theory, was used in [BLR] and also in [GHRU].

There is also work on *differentially private learning algorithms* in which the goal is to give differentially private variants of various learning algorithms [BDMN, KLN+].

7.1.2 Applications (Part I): Releasing Conjunctions

We use the reduction of Theorem 7.1.1 to obtain new data release algorithms "automatically" from learning algorithms that satisfy the theorem's requirements. Here we describe the distribution-free data release algorithms we obtain for approximating conjunction counting queries. These use learning algorithms (which are themselves distribution-free and require only random examples) based on polynomial threshold functions.

Throughout this section we fix the query class under consideration to be conjunctions. We take $\mathcal{U} = \{0,1\}^d$, and a (monotone) conjunction $q \in \mathcal{Q} = \{0,1\}^d$ is satisfied by u iff $\forall i$ s.t. $q_i = 1$ it is also the case that $u_i = 1$. (Our monotone conjunction results extend easily to general non-monotone conjunctions with parameters unchanged.[1]) Our first result is an algorithm for releasing w-way conjunctions:

Theorem 7.1.2 (Distribution-Free Data Release for w-way conjunctions). *There is an ε-differentially private (α, β, γ)-accurate distribution-free data release algorithm, which accesses the query distribution only by sampling, for the class of w-way monotone Boolean conjunction queries. The algorithm has runtime* $\mathrm{poly}(n)$ *on databases of size n provided that*

$$n \geqslant d^{O\left(\sqrt{w \log\left(\frac{w \log d}{\alpha}\right)}\right)} \cdot \widetilde{O}\left(\frac{\log(1/\beta)^3}{\varepsilon \alpha \gamma^2}\right).$$

Since this is a distribution-free data release algorithm that only accesses the query distribution by sampling, we can use the boosting results of [DRV] and obtain a data release algorithm that generates (w.h.p.) a synopsis that is accurate for *all* queries. This increases the running time to $d^w \cdot \mathrm{poly}(n)$ (because the boosting algorithm needs to enumerate over all the w-way conjunctions). The required bound on the database size increases slightly but our big-Oh notation hides this small increase. The corollary is stated formally below:

Corollary 7.1.3 (Boosted Data Release for w-way Conjunctions). *There is an ε-differentially private $(\alpha, \beta, \gamma = 0)$-accurate distribution-free data release algorithm for the class of w-way monotone Boolean conjunction queries with runtime $d^w \cdot \mathrm{poly}(n)$ on databases of size n, provided that*

$$n \geqslant d^{O\left(\sqrt{w \log\left(\frac{w \log d}{\alpha}\right)}\right)} \cdot \widetilde{O}\left(\frac{\log(1/\beta)^3}{\varepsilon \alpha}\right).$$

We also obtain a new data release algorithm for releasing the answers to *all* conjunctions:

Theorem 7.1.4 (Distribution-Free Data Release for *All* Conjunctions). *There is an ε-differentially private (α, β, γ)-accurate distribution-free data release algorithm, which accesses the query distribution only by sampling, for the class of all*

[1] To see this, extend the data domain to be $\{0,1\}^{2d}$, and for each item in the original domain include also its negation. General conjunctions in the original data domain can now be treated as monotone conjunctions in the new data domain. Note that the locality of a conjunction is unchanged. Our results in this section are for arbitrary distributions over the set of monotone conjunctions (over the new domain), and so they will continue to apply to arbitrary distributions on general conjunctions over the original data domain.

monotone Boolean conjunction queries. The algorithm has runtime $\mathrm{poly}(n)$ *on databases of size* n*, provided that*

$$n \geqslant d^{O(d^{1/3}\cdot \log^{2/3}(\frac{d}{\alpha}))} \cdot \widetilde{O}\left(\frac{\log(1/\beta)^3}{\varepsilon\alpha\gamma^2}\right).$$

Again, we can apply boosting to this result; this gives improvements over previous work for a certain range of parameters (roughly $w \in [d^{1/3}, d^{2/3}]$). We omit the details.

Related Work on Releasing Conjunctions. Several past works have considered differentially private data release for conjunctions and w-way conjunctions (also known as marginals and contingency tables). As a corollary of their more general Laplace and Gaussian mechanisms, the work of Dwork *et al.* [DMNS] showed how to release all w-way conjunctions in running time $d^{O(w)}$ provided that the database size is at least $d^{O(w)}$. Barak *et al.* [BCD+] showed how to release *consistent* contingency tables with similar database size bounds. The running time, however, was increased to $\exp(d)$. We note that our data-release algorithms do not guarantee consistency. Gupta et al. [GHRU] gave *distribution-specific* data release algorithm for w-way and for all conjunctions. These algorithms work for the uniform distribution over (k-way or general) conjunctions. The database size bound and running time were (roughly) $d^{\widetilde{O}(1/\alpha^2)}$. For distribution-specific data release on the uniform distribution, the dependence on d in their work is better than our algorithms but the dependence on α is worse. Finally, we note that the general information-theoretic algorithms for differentially private data release also yield algorithms for the specific case of conjunctions. These algorithms are (significantly) more computationally expensive, but they have better database size bounds. For example, the algorithm of [HR] has running time $\exp(d)$ but database size bound is (roughly) $\widetilde{O}(d/\alpha^2)$ (for the relaxed notion of (ε, δ)-differential privacy).

In terms of negative results, Ullman and Vadhan [UV] showed that, under mild cryptographic assumptions, no data release algorithm for conjunctions (even 2-way) can output a *synthetic database* in running time less than $\exp(d)$ (this holds even for *distribution-specific* data release on the uniform distribution). Our results side-step this negative result because the algorithms do not release a synthetic database.

Kasiviswanathan *et al.* [KRSU] showed a lower bound of $\widetilde{\Omega}\left(\min\left\{d^{w/2}/\alpha, 1/\alpha^2\right\}\right)$ on the database size needed for releasing w-way conjunctions. To see that this is consistent with our bounds, note that our bound on n is always larger than $f(\alpha) = 2^{\sqrt{w\log(1/\alpha)}}/\alpha$. We have $f(\alpha) < 1/\alpha^2$ only if $w < \log(1/\alpha)$. But in the range where $w < \log(1/\alpha)$ our theorem needs n to be larger than d^w/α which is consistent with the lower bound.

7.1.3 Applications (Part II): Fourier-Based Approach

We also use Theorem 7.1.1 (in its more general formulation given in Section 7.2.2) to obtain new data release algorithms for answering parity counting queries (in polynomial time) and general AC^0 counting queries (in quasi-polynomial time). For both of these we fix the data universe to be $\mathcal{U} = \{0,1\}^d$, and take the set of query descriptions to also be $\mathcal{Q} = \{0,1\}^d$ (with different semantics for queries in the two cases). Both algorithms are distribution-specific, working for the uniform distribution over query descriptions,[2] and both instantiate the reduction with learning algorithms that use Fourier analysis of the target function. Thus the full data release algorithms use Fourier analysis of the database (viewed as a function on queries).

Parity Counting Queries. Here we consider counting queries that, for a fixed $q \in \{0,1\}^d$, output how many items in the database have inner product 1 with q (inner products are taken over $GF[2]$). I.e., we use the parity predicate $P(q,u) = \sum_i q_i \cdot u_i \pmod 2$. We obtain a polynomial-time data release algorithm for this class (w.r.t. the uniform distribution over queries). This uses our reduction, instantiated with Jackson's Harmonic Sieve learning algorithm [Jac]. In Section 7.5 we prove:

Theorem 7.1.5 (Uniform Distribution Data Release for Parity Counting Queries.). *There is an ε-differentially private algorithm for releasing the class of parity queries over the uniform distribution on $\mathcal{Q}$. For databases of size n, the algorithm has runtime* $\mathrm{poly}(n)$ *and is (α, β, γ)-accurate, provided that*

$$n \geqslant \frac{\mathrm{poly}(d, 1/\alpha, 1/\gamma, \log(1/\beta))}{\varepsilon}.$$

AC^0 Counting Queries. We also consider a quite general class of counting queries, namely, any query family whose predicate is computed by a constant depth (AC^0) circuit. For any family of this type, in Section 7.5 we obtain a data release algorithm over the uniform distribution that requires a database of quasi-polynomial (in d) size (and has running time polynomial in the database size, or quasi-polynomial in d).

Theorem 7.1.6 (Uniform Distribution Data Release for AC^0 Counting Queries). *Take $\mathcal{U} = \mathcal{Q} = \{0,1\}^d$, and $P(q,u) : \mathcal{Q} \times \mathcal{U} \to \{0,1\}$ a predicate computed by a Boolean circuit of depth $\ell = O(1)$ and size* $\mathrm{poly}(d)$. *There is an ε-differentially private data release algorithm for this query class over the uniform distribution*

[2] More generally, we can get results for *smooth* distributions, we defer these to the full version.

on $\mathcal{Q}$. For databases of size n, the algorithm has runtime $\mathrm{poly}(n)$ *and is* (α, β, γ)*-accurate, provided that:*

$$n \geqslant d^{O\left(\log^{\ell}\left(\frac{d}{\alpha\gamma}\right)\right)} \cdot \widetilde{O}\left(\frac{\log^3(1/\beta)}{\varepsilon\alpha^2\gamma}\right).$$

This result uses our reduction instantiated with an algorithm of Jackson *et al.* [JKS] for learning Majority-of-AC^0 circuits. To the best of our knowledge, this is the first positive result for private data release that uses the (circuit) structure of the query class in a "non black-box" way to approximate the query answer. We note that the class of AC^0 predicates is quite rich. For example, it includes conjunctions, approximate counting [Ajt], and $GF[2]$ polynomials with $\mathrm{polylog}(d)$ many terms. While our result is specific to the uniform distribution over $\mathcal{Q}$, we note that some query sets (and query descriptions) may be amenable to *random self-reducibility*, where an algorithm providing accurate answers to uniformly random $q \in Q$ can be used to get (w.h.p.) accurate answers to *any* $q \in Q$. We also note that Theorem 7.1.6 leaves a large degree of freedom in how a class of counting queries is to be represented. Many different sets of query descriptions $\mathcal{Q}$ and predicates $P(q,u)$ can correspond to the same set of counting queries over the same $\mathcal{U}$, and it may well be the case that some representations are more amenable to computations in AC^0 and/or random self-reducibility. Finally, we note that the hardness results of Dwork *et al.* [DNR+] actually considered (and ruled out) efficient data-release algorithms for AC^0 counting queries (even for the uniform distribution case), but only when the algorithm's output is a synthetic database. Theorem 7.1.6 side-steps these negative results because the output is not a synthetic database.

7.1.4 Preliminaries

A class of *counting queries* is specified by a predicate $P\colon \mathcal{Q} \times \mathcal{U} \to \{0,1\}$ where $\mathcal{Q}$ is a set of query descriptions. Each $q \in \mathcal{Q}$ specifies a query and the answer for a query $q \in \mathcal{Q}$ on a single data item $u \in \mathcal{U}$ is given by $P(q,u)$. The answer of a counting query $q \in \mathcal{Q}$ on a data set D is defined as $\frac{1}{n}\sum_{u \in D} P(q,u)$.

We will often fix a data item u and database $D \in \mathcal{D}_n$ and use the following notation:

— $p_u\colon \mathcal{Q} \to \{0,1\}$, $p_u(q) \stackrel{\text{def}}{=} P(q,u)$. The predicate on a fixed data item u.

— $f^D\colon \mathcal{Q} \to [0,1]$, $f^D(q) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{u \in D} P(q,u)$. For an input query description and fixed database, counts the fraction of database items that satisfy that query.

— $f_t^D : \mathcal{Q} \to \{0,1\}$, $f_t^D(q) \stackrel{\text{def}}{=} \mathbb{I}\left\{f^D(q) \geqslant t\right\}$. For an input query description and fixed database and threshold $t \in [0,1]$, indicates whether the fraction of database items that satisfy that query is at least t. Here and in the following $\mathbb{I}$ denotes the 0/1-indicator function.

Special classes of counting queries. We close this section with some concrete examples of query classes that we will consider. Fix $\mathscr{U} = \{0,1\}^d$ and $\mathcal{Q} = \{0,1\}^d$. The query class of *monotone boolean conjunctions* is defined by the predicate $P(q,u) = \bigwedge_{i:\, q_i=1} u_i$. Note that we may equivalently write $P(q,u) = 1 - \bigvee_{i:\, u_i=0} q_i$. The query class of *parities over* $\{0,1\}^d$ is defined by the predicate $P(q,u) = \sum_{i:u_i=1} q_i \pmod 2$.

7.2 Private data release via learning thresholds

In this section we describe our reduction from private data release to a related computational learning task of learning thresholded sums. Section 7.2.1 sets the stage, first introducing definitions for handling distributions and access to an oracle, and then proceeds with notation and formal definitions of (non-interactive) data release and of learning threshold functions. Section 7.2.2 formally states our main theorem giving the reduction, and Section 7.2.3 gives an intuitive overview of the proof. The formal proof is then given in Section 7.3.

7.2.1 Distribution access, data release, learning thresholds

Definition 7.2.1 (Sampling or Evaluation Access to a Distribution). Let G be a distribution over a set $\mathcal{Q}$. When we give an algorithm $\mathscr{A}$ *sampling access* to G, we mean that $\mathscr{A}$ is allowed to sample items distributed by G. When we give an algorithm $\mathscr{A}$ *evaluation access* to G, we mean that $\mathscr{A}$ is both allowed to sample items distributed by G and also to make oracle queries: in such a query $\mathscr{A}$ specifies any $q \in Q$ and receives back the probability $G[q] \in [0,1]$ of q under G. For both types of access we will often measure $\mathscr{A}$'s *sample complexity* or *number of queries* (for the case of evaluation access).[3]

Definition 7.2.2 (Sampling Access to Labeled Examples). Let G be a distribution over a set $\mathcal{Q}$ of potential examples, and let f be a function whose domain is $\mathcal{Q}$. When we give an algorithm $\mathscr{A}$ *sampling access to labeled examples by* (G, f), we mean that $\mathscr{A}$ has sampling access to the distribution $(q, f(q))_{q \sim G}$.

[3] Note that, generally speaking, sampling and evaluation access are incomparably powerful (see [KMR+, Nao]). In this work, however, whenever we give an algorithm evaluation access we will also give it sampling access.

Definition 7.2.3 (Data Release Algorithm)**.** Fix $\mathcal{U}$ to be a data universe, $\mathcal{Q}$ to be a set of query descriptions, $\mathcal{GQ}$ to be a set of distributions on $\mathcal{Q}$, and $P(q,u) : \mathcal{Q} \times \mathcal{U} \to \{0,1\}$ to be a predicate. A $(\mathcal{U},\mathcal{Q},\mathcal{GQ},P)$ *data release algorithm* $\mathscr{A}$ is a (probabilistic) algorithm that gets sampling access to a distribution $G \in \mathcal{GQ}$ and takes as input accuracy parameters $\alpha,\beta,\gamma > 0$, a database size n, and a database $D \in \mathcal{D}_n$. $\mathscr{A}$ outputs a *synopsis* $S : \mathcal{Q} \to [0,1]$.

We say that $\mathscr{A}$ is (α,β,γ)-*accurate* for databases of size n, if for every database $D \in \mathcal{D}_n$ and query distribution $G \in \mathcal{GQ}$:

$$\mathop{\mathbb{P}}_{S \leftarrow \mathscr{A}(n,D,\alpha,\beta,\gamma)} \left\{ \mathop{\mathbb{P}}_{q \sim G} \left\{ |S(q) - f^D(q)| > \alpha \right\} > \gamma \right\} < \beta \tag{7.1}$$

We also consider data release algorithms that get evaluation access to G. In this case, we say that $\mathscr{A}$ is a *data release algorithm using evaluation access*. The definition is unchanged, except that $\mathscr{A}$ gets this additional form of access to G.

When P and $\mathcal{U}$ are understood from the context, we sometimes refer to a $(\mathcal{U},\mathcal{Q},\mathcal{GQ},P)$ data release algorithm as an *algorithm for releasing the class of queries $\mathcal{Q}$ over $\mathcal{GQ}$*.

We note two cases of particular interest. The first is when $\mathcal{GQ}$ is the set of *all distributions* over $\mathcal{Q}$. In this case, we say that $\mathscr{A}$ is a *distribution-free* data release algorithm. For such algorithms it is possible to apply the "boosting for queries" results of [DRV] and obtain a data release algorithm whose synopsis is (w.h.p.) accurate on *all* queries (i.e. with $\gamma = 0$). We note that those boosting results apply only to data release algorithms that access their distribution by sampling (i.e. they need not hold for data release algorithms that use evaluation access). A second case of interest is when $\mathcal{GQ}$ contains only a single distribution, the uniform distribution over all queries $\mathcal{Q}$. In this case both sampling and evaluation access are easy to simulate.

Throughout this work, we fix the accuracy parameter α, and lower bound the required database size n needed to ensure the (additive) approximation error is at most α. See also Remark 2.9.2. Our database size bounds can be converted to error bounds in the natural way.

Definition 7.2.4 (Learning Thresholds)**.** Let $\mathcal{Q}$ be a set (which we now view as a domain of potential unlabeled examples) and let $\mathcal{GQ}$ be a set of distributions on $\mathcal{Q}$. Let $\mathcal{F}$ be a set of predicates on $\mathcal{Q}$, i.e. functions $Q \to \{0,1\}$. Given $t \in [0,1]$, let $\mathcal{F}_{n,t}$ be the set of all threshold functions of the form $f = \mathbb{I}\left\{\frac{1}{n}\sum_{i=1}^{n} f_i \geqslant t\right\}$ where $f_i \in \mathcal{F}$ for all $1 \leqslant i \leqslant n$. We refer to functions in $\mathcal{F}_{n,t}$ as *n-thresholds over $\mathcal{F}$*. Let $\mathscr{L}$ be a (probabilistic) algorithm that gets sampling access to labeled examples by a distribution $G \in \mathcal{GQ}$ and a target

function $f \in \mathcal{F}_{n,t}$. $\mathscr{L}$ takes as input accuracy parameters $\gamma, \beta > 0$, an integer $n > 0$, and a threshold $t \in [0,1]$. $\mathscr{L}$ outputs a boolean *hypothesis* $h : \mathcal{Q} \to \{0,1\}$.

We say that $\mathscr{L}$ is an (γ,β)-*learning algorithm for thresholds over* $(\mathcal{Q}, \mathcal{GQ}, \mathcal{F})$ if for every $\gamma, \beta > 0$, every n, every $t \in [0,1]$, every $f \in \mathcal{F}_{n,t}$ and every $G \in \mathcal{GQ}$, we have

$$\mathop{\mathbb{P}}_{h \leftarrow \mathscr{L}(n,t,\gamma,\beta)} \left\{ \mathop{\mathbb{P}}_{q \sim G} \{h(q) \neq f(q)\} > \gamma \right\} < \beta. \tag{7.2}$$

The definition is analogous for all other notions of oracle access (see e.g. Definition 7.2.5 below).

7.2.2 Statement of the main theorem

In this section we formally state our main theorem, which establishes a general reduction from private data release to learning certain threshold functions. The next definition captures a notion of oracle access for learning algorithms which arises in the reduction. The definition combines sampling access to labeled examples with a limited kind of evaluation access to the underlying distribution and black-box oracle access to the target function f.

Definition 7.2.5 (approximate distribution-restricted oracle access)**.** Let G be a distribution over a domain $\mathcal{Q}$, and let f be a function whose domain is $\mathcal{Q}$. When we say that an algorithm $\mathscr{A}$ has *approximate G-restricted evaluation access to f*, we mean that

1. $\mathscr{A}$ has sampling access to labeled examples by (G, f); and
2. $\mathscr{A}$ can make oracle queries on any $q \in Q$, which are answered as follows: there is a fixed constant $c \in [1/3, 3]$ such that (i) if $G[q] = 0$ the answer is $(0, \perp)$; and (ii) if $G[q] > 0$ the answer is a pair $(c \cdot G[q], f(q))$.

Remark 7.2.6. We remark that this is the type of of oracle access provided to the learning algorithm in our reduction. This is different from the oracle access that the data release algorithm has. We could extend Definition 7.2.3 to refer to approximate evaluation access to G; all our results on data release using evaluation access would extend to this weaker access (under appropriate approximation guarantees). For simplicity, we focus on the case where the *data release algorithm* has perfectly accurate evaluation access, since this is sufficient throughout for our purpose.

One might initially hope that privately releasing a class of queries $\mathcal{Q}$ over some set of distributions $\mathcal{GQ}$ reduces to learning corresponding threshold functions over the *same* set of distributions. However, our reduction will need a learning algorithm that works for a potentially larger set of distributions

$\mathcal{GQ}' \supseteq \mathcal{GQ}$. (We will see in Theorem 7.2.8 that this poses a stronger requirement on the learning algorithm.) Specifically, $\mathcal{GQ}'$ will be a *smooth extension* of $\mathcal{GQ}$ as defined next.

Definition 7.2.7 (smooth extensions)**.** Given a distribution G over a set $\mathcal{Q}$ and a value $\mu \geqslant 1$, the *μ-smooth extension of G* is the set of all distributions G' which are such that $G'[q] \leqslant \mu \cdot G[q]$ for all $q \in \mathcal{Q}$. Given a set of distributions $\mathcal{GQ}$ and $\mu \geqslant 1$, the *μ-smooth extension of $\mathcal{GQ}$*, denoted $\mathcal{GQ}'$, is defined as the set of all distributions that are a μ-smooth extension of some $G \in \mathcal{GQ}$.

With these two definitions at hand, we can state our reduction in its most general form. We will combine this general reduction with specific learning results to obtain concrete new data release algorithms in Sections 7.4 and 7.5.

Theorem 7.2.8 (Main Result: Private Data Release via Learning Thresholds)**.** *Let $\mathcal{U}$ be a data universe, $\mathcal{Q}$ a set of query descriptions, $\mathcal{GQ}$ a set of distributions over $\mathcal{Q}$, and $P\colon \mathcal{Q} \times \mathcal{U} \to \{0,1\}$ a predicate.*

Then, there is an ε-differentially private (α, β, γ)-accurate data-release algorithm for databases of size n provided that

- *there is an algorithm $\mathscr{L}$ that (γ,β)-learns thresholds over $(\mathcal{Q}, \mathcal{GQ}', \{p_u : u \in \mathcal{U}\})$, running in time $t(n,\gamma,\beta)$ and using $b(n,\gamma,\beta)$ queries to an approximate distribution-restricted evaluation oracle for the target n-threshold function, where $\mathcal{GQ}'$ is the $(2/\gamma)$-smooth extension of $\mathcal{GQ}$; and*
- *we have*

$$n \geqslant \frac{C \cdot b(n',\gamma',\beta') \cdot \log\left(\frac{b(n',\gamma',\beta')}{\alpha\gamma\beta}\right) \cdot \log(1/\beta')}{\varepsilon\alpha^2\gamma}, \tag{7.3}$$

  where $n' = \Theta(\log|\mathcal{Q}|/\alpha^2)$, $\beta' = \Theta(\beta\alpha)$, $\gamma' = \Theta(\gamma\alpha)$ and $C > 0$ is a sufficiently large constant.

The running time of the data release algorithm is $\mathrm{poly}(t(n',\gamma',\beta'), n, 1/\alpha, \log(1/\beta), 1/\gamma)$.

The next remark points out two simple modifications of this theorem.

Remark 7.2.9.

1. We can improve the dependence on n in (7.3) by a factor of $\Theta(1/\alpha)$ in the case where the learning algorithm $\mathscr{L}$ only uses sampling access to labeled examples. In this case the data release algorithm also uses only sampling access to the query distribution G. The precise statement is given in Theorem 7.3.10 which we present after the proof of Theorem 7.2.8.

2. A similar theorem holds for (ε,δ)-differential privacy, where the requirement on n in (7.3) is improved to a requirement on $\sqrt{n}$ up to a $\log(1/\delta)$ factor. The proof is the same, except for a different (but standard) privacy argument, e.g., using the composition theorem we saw in Section 2.4.

7.2.3 Informal proof overview

Our goal in the data release setting is approximating the query answers $\{f^D(q)\}_{q\in Q}$. This is exactly the task of approximating or *learning* a sum of n predicates from the set $\mathcal{F} = \{p_u : u \in \mathscr{U}\}$. Indeed, each item u in the database specifies a predicate p_u, and for a fixed query $q \in Q$ we are trying to approximate the sum of the predicates $f^D(q) = \frac{1}{|D|} \cdot \sum_{u\in D} p_u(q)$. We want to approximate such a sum in a privacy-preserving manner, and so we will only permit limited access to the function f^D that we try to approximate. In particular, we will only allow a bounded number of noisy oracle queries to this function. Using standard techniques (i.e. adding appropriately scaled Laplace noise [DMNS]), an approximation obtained from a bounded number of noisy oracle queries will be differentially private. It remains, then, to tackle the task of (i) learning a sum of n predicates from $\mathcal{F}$ using an oracle to the sum, and (ii) doing so using only a *bounded (smaller than n) number* of oracle queries when we are provided *noisy answers*.

From Sums to Thresholds. Ignoring privacy concerns, it is straightforward to reduce the task of learning a sum f^D of predicates (given an oracle for f^D) to the task of learning thresholded sums of predicates (again given an oracle for f^D). Indeed, set $k = \lceil 3/\alpha \rceil$ and consider the thresholds $t_1, \ldots, t_k$ given by $t_i = i/(k+1)$. Now, given an oracle for f^D, it is easy to simulate an oracle for $f^D_{t_i}$ for any t_i. Thus, we can learn each of the threshold functions $f^D_{t_i}$ to accuracy $1 - \gamma/k$ with respect to G. Call the resulting hypotheses $h_1, \ldots, h_k$. Each h_i labels a $(1-\gamma/k)$-fraction of the queries/examples in Q correctly w.r.t the threshold function $f^D_{t_i}$. We can produce an aggregated hypothesis h for approximating f^D as follows: given a query/example q, let $h(q)$ equal t_i where t_i is the smallest i such that $h_i(q) = 0$ and $h_{i+1}(q) = 1$. For random $q \sim G$, we will then have $|h(q) - f^D(q)| \leqslant \alpha/3$ with probability $1 - \gamma$ (over the choice of q).

Thus, we have reduced learning a sum to learning thresholded sums (where in both cases the learning is done with an oracle for the sum). But because of privacy considerations, we must address the challenges mentioned above: (i) learning a *thresholded* sum of n predicates using few (less than n) oracle queries to the sum, and (ii) learning when the oracle for the sum can return noisy answers. In particular, the noisy sum answers can induce errors on threshold oracle queries (when the sum is close to the threshold).

Restricting to Large Margins. Let us say that a query/example $q \in \mathcal{Q}$ has *low margin with respect to f^D and t_i* if $|f^D(q) - t_i| \leqslant \alpha/7$. A useful observation is that in the argument sketched above, we do *not* need to approximate each threshold function $f^D_{t_i}$ well on low margin elements q. Indeed, suppose that

each hypothesis h_i errs arbitrarily on a set $E_i \subseteq \mathcal{Q}$ that contains only inputs that have low margin w.r.t. f^D and t_i, but achieves high accuracy $1 - \gamma/k$ with respect to G conditioned on the event $\mathcal{Q} \setminus E_i$. Then the above aggregated hypothesis h would still have high accuracy with high probability over $q \sim G$; more precisely, h would satisfy $|h(q) - f^D(q)| \leqslant 2\alpha/3$ with probability $1 - \gamma$ for $q \sim G$.

The reason is that for every $q \in Q$, there can only be one threshold $i^* \in \{1, \ldots, k\}$ such that $|f^D(q) - t_{i^*}| \leqslant \alpha/7$ (since any two thresholds are $\alpha/3$- apart from each other). While the threshold hypothesis h_{i^*} might err on q (because q has low margin w.r.t. t_{i^*}), the hypotheses h_{i^*-1} and h_{i^*+1} should still be accurate (w.h.p. over $q \sim G$), and thus the aggregated hypothesis h will still output a value between t_{i^*-1} and t_{i^*+1}.

Threshold Access to The Data Set. We will use the above observation to our advantage. Specifically, we restrict all access to the function f^D to what we call a *threshold oracle*. Roughly speaking, the threshold oracle (which we denote $\mathcal{TO}$ and define formally in Section 7.3.1) works as follows: when given a query q and a threshold t, it draws a suitably scaled Laplacian variable N (used to ensure differential privacy) and returns 1 if $f^D(q) + N \geqslant t + \alpha/20$; returns 0 if $f^D(q) + N \leqslant t - \alpha/20$; and returns "$\perp$" if $t - \alpha/20 < f^D(q) + N < t + \alpha/20$. If D is large enough then we can ensure that $|N| \leqslant \alpha/40$ with high probability, and thus whenever the oracle outputs $\perp$ on a query q we know that q has low margin with respect to f^D and t (since $\alpha/20 + |N| < \alpha/7$).

We will run the learning algorithm $\mathscr{L}$ on examples generated using the oracle $\mathcal{TO}$ after removing all examples for which the oracle returned $\perp$. Since we are conditioning on the $\mathcal{TO}$ oracle not returning $\perp$, this transforms the distribution G into a conditional distribution which we denote G'. Since we have only conditioned on removing low-margin q's, the argument sketched above applies. That is, the hypothesis that has high accuracy with respect to this conditional distribution G' is still useful for us.

So the threshold oracle lets us use noisy sum answers (allowing the addition of noise and differential privacy), but in fact it also addresses the second challenge of reducing the query complexity of the learning algorithm. This is described next.

Savings in Query Complexity via Subsampling. The remaining challenge is that the threshold oracle can be invoked only (at most) n times before we exceed our "privacy budget". This is problematic, because the query complexity of the underlying learning algorithm may well depend on n, since f^D is a sum of n predicates. To reduce the number of oracle queries that need to be made, we observe that the sum of n predicates that we are trying to learn

can actually be approximated by a sum of fewer predicates. In fact, there exists a sum $f^{D'}$ of $n' = O(\log|\mathcal{Q}|/\alpha^2)$ predicates from $\mathcal{F}$ that is $\alpha/100$-close to f^D on all inputs in $\mathcal{Q}$, i.e. $|f^D(q) - f^{D'}(q)| \leqslant \alpha/100$ for all $q \in \mathcal{Q}$. (The proof is by a subsampling argument, as in [BLR]; see Section 7.3.1.) We will aim to learn this "smaller" sum. The hope is that the query complexity for learning $f^{D'}$ may be considerably smaller, namely scaling with n' rather than n. Notice, however, that learning a threshold of $f^{D'}$ requires a threshold oracle to $f^{D'}$, rather than the threshold oracle we have, which is to f^D. Our goal, then, is to use the threshold oracle to f^D to simulate a threshold oracle to $f^{D'}$. This will give us "the best of both worlds": we can make (roughly) $O(n)$ oracle queries thus preserving differential privacy, while using a learning algorithm that is allowed to have query complexity superlinear in n'.

The key observation showing that this is indeed possible is that the threshold oracle $\mathcal{TO}$ already "avoids" low-margin queries where f_t^D and $f_t^{D'}$ might disagree! Whenever the threshold oracle $\mathcal{TO}$ (w.r.t. D) answers $l \neq \perp$ on a query q,, we must have $|f^D(q) - t| \geqslant \alpha/20 - N > \alpha/100$, and thus $f_t^D(q) = f_t^{D'}(q)$. Moreover, it is still the case that $\mathcal{TO}$ only answers $\perp$ on queries q that have low margins w.r.t $f_t^{D'}$. This means that, as above, we can run $\mathscr{L}$ using $\mathcal{TO}$ (w.r.t. D) in order to learn $f^{D'}$. The query complexity depends on n' and is therefore independent of n. At the same time, we continue to answer all queries using the threshold oracle with respect to f^D so that our privacy budget remains on the order $|D| = n$. Denoting the query complexity of the learning algorithm by $b(n')$ we only need that $n \gg b(n')$. This allows us to use learning algorithms that have $b(n') \gg n'$ as is usually the case.

Sampling from the conditional distribution. In the exposition above we glossed over one technical detail, which is that the learning algorithm requires sampling (or distribution restricted) access to the distribution G' over queries q on which $\mathcal{TO}$ does not return $\perp$, whereas the data release algorithm we are trying to build only has access to the original distribution G. We reconcile this disparity as follows.

For a threshold t, let ζ_t denote the probability that the oracle $\mathcal{TO}$ does not return $\perp$ when given a random $q \sim G$ and the threshold t. There are two cases depending on ζ_t:

$\zeta_t < \gamma$: This means that the threshold t is such that with probability $1 - \gamma$ a random sample $q \sim G$ has low margin with respect to f^D and t. In this case, by simply outputting the constant-t function as our approximation for f^D, we get a hypothesis that has accuracy $\alpha/3$ with probability $1 - \gamma$ over random $q \sim G$.

$\zeta_t \geqslant \gamma$: In this case, the conditional distribution G' induced by the threshold

oracle is $1/\gamma$-smooth w.r.t. G. In particular, G' is contained in the smooth extension $\mathcal{GQ}'$ for which the learning algorithm is guaranteed to work (by the conditions of Theorem 7.2.8). This means that it we can sample from G' using rejection sampling to G. It suffices to oversample by a factor of $O(1/\gamma)$ to make sure that we receive enough examples that are not rejected by the threshold oracle.

Finally using a reasonably accurate estimate of ζ, we can also implement the distribution restricted approximate oracle access that may be required by the learning algorithm. We omit the details from this informal overview.

7.3 Proof of the main theorem

In this section, we give a formal proof of Theorem 7.2.8. We formalize and analyze the threshold oracle first. Then we proceed to our main reduction.

7.3.1 Threshold access and subsampling

We begin by describing the threshold oracle that we use to access the function f^D throughout our reduction; it is presented in Figure 7.1. The oracle has two purposes. One is to ensure differential privacy by adding noise every time we access f^D. The other purpose is to "filter out" queries that are too close to the given threshold. This will enable us to argue that the threshold oracle for f_t^D agrees with the function $f_t^{D'}$ where D' is a small subsample of D.

Throughout the remainder of this section we fix all input parameters to our oracle, i.e. the data set D and the values $b, \alpha > 0$. We let $\beta > 0$ denote the desired error probability of our algorithm.

Lemma 7.3.1. *Call two queries $(q,t),(q',t')$* distinct *if $q \neq q'$. Then, the threshold oracle $\mathcal{TO}(D,\alpha,b)$ answers any sequence of b distinct adaptive queries to f^D with ε-differential privacy.*

Proof. This follows directly from the guarantees of the Laplacian mechanism as shown in [DMNS]. ∎

Our goal is to use the threshold oracle for f_t^D to correctly answer queries to the function $f_t^{D'}$ where D' is a smaller (sub-sampled) database that gives "close" answers to D on all queries $q \in \mathcal{Q}$. The next lemma shows that there always exists such a smaller database.

Lemma 7.3.2. *For any $\alpha \geqslant 0$, there is a database D' of size*

$$|D'| \leqslant \frac{10 \log |\mathcal{Q}|}{\alpha^2} \tag{7.4}$$

Input: data set D of size n, tolerance $\alpha > 0$, query bound $b \in \mathbb{N}$.
Threshold Oracle $\mathcal{TO}(D, \alpha, b)$:

— When invoked on the j-th query $(q, t) \in \mathcal{Q} \times [0, 1)$, do the following:

– If $j > b$, output $\perp$ and terminate.

– If (q, t') has not been asked before for any threshold t', sample a fresh Laplacian variable $N_q \sim Lap(b/\varepsilon n)$ and put $A_q = f^D(q) + N_q$. Otherwise reuse the previously created value A_q.

–

$$\text{Output} \begin{cases} 0 & \text{if } A_q \leqslant t - 2\alpha/3, \\ 1 & \text{if } A_q \geqslant t + 2\alpha/3, \\ \perp & \text{otherwise.} \end{cases}$$

Figure 7.1: Threshold oracle for f^D. This threshold oracle is the only way in which the data release algorithm ever interacts with the data set D. Its purpose is to ensure privacy and to reject queries that are too close to a given threshold.

such that

$$\max_{q \in \mathcal{Q}} \left| f^D(q) - f^{D'}(q) \right| < \alpha \,.$$

Proof. The existence of D' follows from a subsampling argument as shown in [BLR]. ∎

The next lemma states the two main properties of the threshold oracle that we need. To state them more succinctly, let us denote by

$$Q(t, \alpha) = \{q \in \mathcal{Q} \colon |f^D(q) - t| \geqslant \alpha\}$$

the set of elements in $\mathcal{Q}$ that are α-far from the threshold t.

Lemma 7.3.3 (Agreement). *Suppose D satisfies*

$$|D| \geqslant \frac{30b \cdot \log(b/\beta)}{\varepsilon \alpha}, \tag{7.5}$$

Then, there is a data set D' of size $|D'| \leqslant 90 \cdot \alpha^{-2} \log|\mathcal{Q}|$ and an event Γ (only depending on the choice of the Laplacian variables) such that Γ has probability $1 - \beta$ and if Γ occurs, then $\mathcal{TO}(D, \alpha, b)$ has the following guarantee: whenever $\mathcal{TO}(D, \alpha, b)$ outputs l on one of the queries (q, t) in the sequence, then

1. *if $l \neq \perp$ then $l = f_t^{D'}(q) = f_t^D(q)$, and*
2. *if $l = \perp$ then $q \notin Q(t, \alpha)$.*

Proof. Let D' be the data set given by Lemma 7.3.2 with its "α" value set to $\alpha/3$ so that

$$\left|f^D(q) - f^{D'}(q)\right| < \alpha/3$$

for every input $q \in \mathcal{Q}$.

The event Γ is defined as the event that every Laplacian variable N_q sampled by the oracle has magnitude $|N_q| < \alpha/3$. Under the given assumption on $|D|$ in 7.5 and using basic tail bounds for the Laplacian distribution, this happens with probability $1 - \beta$.

Assuming Γ occurs, the following two statements hold:

1. Whenever the oracle outputs $l \neq \perp$ on a query (q, t), then we must have either $f^D(q) + N_q - t \geqslant 2\alpha/3$ (and thus both $f^D(q) > t + \alpha/3$ and $f^{D'}(q) > t$) or else $f^D(q) + N_q - t \leqslant -2\alpha/3$ (and thus both $f^D(q) < t - \alpha/3$ and $f^{D'}(q) < t$). This proves the first claim of the lemma.

2. Whenever $q \in Q(t, \alpha)$, then $|f^D(q) + N_q - t| \geqslant 2\alpha/3$, and therefore the oracle does not output $\perp$. This proves the second claim of the lemma.

■

7.3.2 Privacy-preserving reduction

In this section we describe how to convert a non-private learning algorithm for threshold functions of the form f_t^D to a privacy-preserving learning algorithm for functions of the form f^D. The reduction is presented in Figure 7.2. We call the algorithm PrivLearn.

Setting of parameters. In the description of PrivLearn we use the following setting of parameters:

$$n' = \frac{4410 \cdot \log|\mathcal{Q}|}{\alpha^2} \qquad k = \left\lceil \frac{3}{\alpha} \right\rceil \qquad \gamma' = \frac{\gamma}{k} \qquad \beta' = \frac{\beta}{6k} \tag{7.6}$$

$$b_{\text{base}} = b(n', \gamma', \beta') \qquad b_{\text{iter}} = \frac{100 b_{\text{base}} \cdot \log(1/\beta')}{\gamma} \qquad b_{\text{total}} = 2k \cdot b_{\text{iter}} \tag{7.7}$$

Analysis of the reduction. Throughout the analysis of the algorithm we keep all input parameters fixed so as to satisfy the assumptions of Theorem 7.2.8. Specifically we will need

$$|D| \geqslant \frac{210 \cdot b_{\text{total}} \cdot \log(10 b_{\text{total}}/\beta)}{\varepsilon\alpha}. \tag{7.8}$$

We have made no attempt to optimize various constants throughout.

Input: Distribution $G \in \mathcal{GQ}$, data set D of size n, accuracy parameters $\alpha, \beta, \gamma > 0$; learning algorithm $\mathscr{L}$ for thresholds over $(\mathcal{Q}, \mathcal{GQ}, \mathcal{F})$ as in Theorem 7.2.8 requiring $b(n', \gamma', \beta')$ labeled examples and approximate restricted evaluation access to the target function.
Parameters: See (7.6) and (7.7).
Algorithm PrivLearn for privately learning f^D:

1. Let $\mathcal{TO}$ denote an instantiation of $\mathcal{TO}(D, \alpha/7, b_{\text{total}})$.
2. Sample b_{iter} points $\{q_j\}_{1 \leqslant j \leqslant b_{\text{iter}}}$ from G.
3. For each iteration $i \in \{1, \ldots, k\}$:
   (a) Let $t_i = i/k + 1$.
   (b) For each q_j, $j \in [b_{\text{iter}}]$ send the query (q_j, t_i) to $\mathcal{TO}$ and let l_j denote the answer. Let $B_i = \{j : l_j \neq \perp\}$.
   (c) If $\frac{|B_i|}{b_{\text{iter}}} < \frac{\gamma}{2}$, output the constant t_i function as hypothesis h and terminate the algorithm.
   (d) Run the learning algorithm $\mathscr{L}(n', t_i, \gamma', \beta')$ on the labeled examples $\{(q_j, l_j)\}_{j \in B_i}$, answering evaluation queries from $\mathscr{L}$ as follows:
      — Given a query q posed by $\mathscr{L}$, let l be the answer of $\mathcal{TO}$ on (q, t_i).
      — If $l = \perp$, then output $(0, \perp)$. Otherwise, output $(G[q] \cdot \frac{b_{\text{iter}}}{|B_i|}, l)$.
   (e) Let h_i denote the resulting hypothesis.
4. Having obtained hypotheses $h_1, \ldots, h_k$, the final hypothesis h is defined as follows: $h(q)$ equals the smallest $i \in [k]$ such that $h_i(q) = 1$ and $h_{i-1}(q) = 0$ (we take $h_0(q) = 0$ and $h_{k+1}(q) = 1$).

Figure 7.2: Reduction from private data release to learning thresholds (non-privately).

Lemma 7.3.4 (Privacy). *Algorithm* PrivLearn *satisfies ε-differential privacy.*

Proof. In each iteration of the loop in Step 3 the algorithm makes at most $2b_{\text{iter}}$ queries to $\mathcal{TO}$ (there are b_{iter} calls made on the samples and at most $b_{\text{base}} \leqslant b_{\text{iter}}$ evaluation queries). But note that $\mathcal{TO}$ is instantiated with a query bound of $b_{\text{total}} = 2kb_{\text{iter}}$. Hence, it follows from Lemma 7.3.1 that $\mathcal{TO}$ satisfies ε-differential privacy. Since $\mathcal{TO}$ is the only way in which PrivLearn ever interacts with the data set, PrivLearn satisfies ε-differential privacy. ∎

We now prove that the hypothesis produced by the algorithm is indeed

accurate, as formalized by the following lemma.

Lemma 7.3.5 (Accuracy). *With overall probability* $1-\beta$*, the hypothesis* h *returned by* PrivLearn *satisfies*

$$\mathop{\mathbb{P}}_{q\sim G}\left\{\left|h(q)-f^D(q)\right|\leqslant\alpha\right\}\geqslant 1-\gamma\,. \tag{7.9}$$

Proof. We consider three possible cases:

1. The first case is that there exists a $t\in\{t_1,\ldots,t_k\}$ such that distribution G has at least $1-\gamma/10$ of its mass on points that are α-close to t. In this case a Chernoff bound and the choice of $b_{\text{iter}}\gg b_{\text{base}}$ imply that with probability $1-\beta$ the algorithm terminates prematurely and the resulting hypothesis satisfies (7.9).

2. In the second case, there exists a $t\in\{t_1,\ldots,t_k\}$ such that the probability mass G puts on points that are α-close to t is between $1-\gamma$ and $1-\gamma/10$. In this case if the algorithm terminates prematurely then (7.9) is satisfied; below we analyze what happens assuming the algorithm does not terminate prematurely.

3. In the third case every $t\in\{t_1,\ldots,t_k\}$ is such that G puts less than $1-\gamma$ of its mass on points α-close to t. In this third case if the algorithm terminates prematurely then (7.9) will not hold; however, our choice of b_{iter} implies that in this third case the algorithm terminates prematurely with probability at most $1-\beta$. As in the second case, below we will analyze what happens assuming the algorithm does not terminate prematurely.

Thus in the remainder of the argument we may assume without loss of generality that the algorithm does not terminate prematurely, i.e. it produces a full sequence of hypotheses $h_1,\ldots,h_k$. Furthermore, we can assume that the distribution G places at most $1-\gamma/10$ fraction of its weight near any particular threshold t_i. This leads to the following claim, showing that in all iterations, the number of labeled examples in B_i is large enough to run the learning algorithm.

Claim 7.3.6. $\mathbb{P}\{\forall i\colon |B_i|\geqslant b_{\text{base}}\}\geqslant 1-\frac{\beta}{3}$.

Proof. By our assumption, the probability that a sample $q\sim G$ is rejected at step t of PrivLearn is at most $\gamma/10$. By the choice of b_{iter} it follows that $|B_i|\geqslant b_{\text{base}}$ with probability $1-\beta/k$. Taking a union bound over all thresholds t completes the proof. ∎

The proof strategy from here on is to first analyze the algorithm on the conditional distribution that is induced by the threshold oracle. We will then pass from this conditional distribution to the actual distribution that we are interested in, namely, G.

We chose $|D|$ large enough so that we can apply Lemma 7.3.3 to $\mathcal{TO}$ with the "α"-setting of Lemma 7.3.3 set to $\alpha/7$. Let D' be the data set and Γ be the event given in the conclusion of Lemma 7.3.3 applied to $\mathcal{TO}$. (Note that $n' = |D'| \leqslant 7^2 \cdot 90\alpha^{-2} \log|\mathcal{Q}|$ as stated above.)

By the choice of our parameters, we have

$$\mathbb{P}\{\Gamma\} \geqslant 1 - \frac{\beta}{3}. \tag{7.10}$$

Here the probability is computed only over the internal randomness of the threshold oracle $\mathcal{TO}$ which we denote by R. Fix the randomness R of $\mathcal{TO}$ such that $R \in \Gamma$. For the sake of analysis, we can think of the randomness of the oracle as a collection of independent random variables $(N_q)_{q\in\mathcal{Q}}$ (where N_q is used to answer all queries of the form (q, t')). In particular, the behavior of the oracle would not change if we were to sample all variables $(N_q)_{q\in\mathcal{Q}}$ up front. When we fix R we thus mean that we fix N_q for all $q \in \mathcal{Q}$.

We may therefore assume for the remainder of the analysis that $\mathcal{TO}$ satisfies properties (1) and (2) of Lemma 7.3.3.

Let us denote by $Q_i \subseteq \mathcal{Q}$ the set of examples for which $\mathcal{TO}$ would not answer $\perp$ in Step 3 at the i-th iteration of the algorithm. Note that this is a well-defined set since we fixed the randomness of the oracle. Denote by G_i the distribution G conditioned on Q_i. Further, let $Z_i = \mathbb{P}_{q\sim G}\{q \in Q_i\}$. Observe that

$$G_i[q] = \begin{cases} G[q]/Z_i & q \in Q_i \\ 0 & \text{o.w.} \end{cases}. \tag{7.11}$$

The next lemma shows that PrivLearn answers evaluation queries with the desired multiplicative precision.

Lemma 7.3.7. *With probability* $1 - \beta/6k$ *(over the randomness of* PrivLearn*), we have*

$$\frac{Z_i}{3} \leqslant \frac{|B_i|}{b_{\text{iter}}} \leqslant 3Z_i. \tag{7.12}$$

Proof. The lemma follows from a Chernoff bound with the fact that we chose $b_{\text{iter}} \gg b_{\text{base}}$. ∎

Assuming that (7.12) holds, we can argue that the learning algorithm in step t produces a "good" hypothesis as expressed in the next lemma.

Lemma 7.3.8. *Let $t \in \{t_1, \ldots, t_k\}$. Conditioned on (7.12), we have that with probability $1 - \beta/6k$ (over the internal randomness of the learning algorithm invoked at step i) the hypothesis h_i satisfies*

$$\mathbb{P}_{q \sim G_i}\left\{h_i(q) = f_{t_i}^{D}(q)\right\} \geqslant 1 - \frac{\gamma}{k}.$$

Proof. This follows directly from the guarantee of the learning algorithm $\mathscr{L}$ once we argue that (with the claimed probability):

1. Each example q is sampled from G_i and labeled correctly by $f_{t_i}^{D'}(q)$ and $f_{t_i}^{D'}(q) = f_{t_i}^{D}(q)$.
2. All evaluation queries asked by the learning algorithm are answered with the multiplicative error allowed in Definition 7.2.5.
3. The algorithm received sufficiently many, i.e., b_{base}, labeled examples.

The first claim follows from the definition of G_i, since we can sample from G_i by sampling from G and rejecting if the oracle $\mathcal{TO}$ returns $\perp$. Since Γ is assumed to hold, we can invoke property (1) of Lemma 7.3.3 to conclude that whenever the oracle does not return $\perp$, then its answer agrees with $f_{t_i}^{D'}(q)$ and moreover $f_{t_i}^{D'}(q) = f_{t_i}^{D}(q)$.

To see the second claim, consider an evaluation query q. We consider two cases. The first case is where the threshold oracle returns $\perp$ and PrivLearn outputs $(0, \perp)$. Note that in this case indeed G_i puts 0 weight on the query q. In the second case PrivLearn outputs $(G[q] \cdot b_{\text{iter}}/|B_i|, l)$. By (7.11) and since we assumed Γ holds, the output satisfies the desired multiplicative bound.

The third claim is a direct consequence of Claim 7.3.6. ∎

We conclude from the above that with probability $1 - \beta/3$ (over the combined randomness of PrivLearn and of the learning algorithm), simultaneously for all $i \in [k]$ we have

$$\mathbb{P}_{q \sim G}\left\{h_i(q) \neq f_{t_i}^{D}(q) \;\middle|\; Q_i\right\} = \mathbb{P}_{q \sim G_i}\left\{h_i(q) \neq f_{t_i}^{D}(q)\right\} \leqslant \frac{\gamma}{k}. \tag{7.13}$$

This follows from a union bound over all k applications of Lemma 7.3.7 and Lemma 7.3.8.

We can now complete the proof of Lemma 7.3.5. That is, we will show that assuming (7.13) the hypothesis h satisfies

$$\mathbb{P}_{q \sim G}\left\{\left|h(q) - f^{D}(q)\right| \leqslant \alpha\right\} \geqslant 1 - \gamma.$$

Note that

1. (7.13) occurs with probability $1 - \beta/3$,
2. our assumption on the threshold oracle, i.e., $R \in \Gamma$ also occurs with probability $1 - \beta/3$ (over the randomness of the oracle)
3. the event in Claim 7.3.6 holds with probability $1 - \beta/3$.

Hence all three events occur simultaneously with probability $1 - \beta$ which is what we claimed. We proceed by assuming that all three events occurred. In the following, let

$$\mathrm{Err}_i = \{q \in \mathcal{Q} \colon h_i(q) \neq f_{t_i}^D(q)\}$$

denote the set of points on which h_i errs. We will need the following claim.

Claim 7.3.9. *Let $q \in \mathcal{Q}$. Then,*

$$\left|h(q) - f^D(q)\right| > \alpha \qquad \Longrightarrow \qquad q \in \bigcup_{i \in [k]} \mathrm{Err}_i \cap Q_i \,.$$

Proof. Arguing in the contrapositive, suppose $q \notin \bigcup_{i\in[k]} \mathrm{Err}_i \cap Q_i$. This means that for all $i \in [k]$ we have that either $q \notin \mathrm{Err}_i$ or $q \notin Q_i$.

However, we claim that there can be at most one $i \in [k]$ such that $q \notin Q_i$ meaning that q is rejected at step i. This follows from property (2) of Lemma 7.3.3 which asserts that if $q \notin Q_i$, then we must have $|f^D(q) - t_i| < \alpha/7$, and the fact that any two thresholds differ by at least $\alpha/3$.

Hence, under the assumption above it must be the case that $q \notin \mathrm{Err}_i$ for all but at most one $i \in [k]$. This means that all but one hypothesis h_i correctly classify q. Since the thresholds are spaced $\alpha/3$ apart, this means the hypothesis h has error at most $2\alpha/3 \leqslant \alpha$ on q. ∎

With the previous claim, we can finish the proof. Indeed,

$$\begin{aligned}
\mathop{\mathbb{P}}_{q\sim G}\left\{\left|h(q) - f^D(q)\right| > \alpha\right\} &\leqslant \mathop{\mathbb{P}}_{q\sim G}\left\{\bigcup_{i\in[k]} \mathrm{Err}_i \cap Q_i\right\} && \text{(using Claim 7.3.9)} \\
&\leqslant \sum_{i=1}^{k} \mathop{\mathbb{P}}_{q\sim G}\{\mathrm{Err}_i \cap Q_i\} && \text{(union bound)} \\
&= \sum_{i=1}^{k} \mathop{\mathbb{P}}_{q\sim G}\{q \in \mathrm{Err}_i \mid Q_i\} \mathop{\mathbb{P}}_{q\sim G}\{Q_i\} \\
&\leqslant \sum_{i=1}^{k} \mathop{\mathbb{P}}_{q\sim G}\{q \in \mathrm{Err}_i \mid Q_i\} \\
&\leqslant k \cdot \frac{\gamma}{k} && \text{(using (7.13))} \\
&= \gamma \,.
\end{aligned}$$

This concludes the proof of Lemma 7.3.5 ■

Lemma 7.3.4 (Privacy) together with Lemma 7.3.5 (Accuracy) conclude the proof of out main theorem, Theorem 7.2.8.

7.3.3 Quantitative improvements without membership queries

Here we show how to shave off a factor of $1/\alpha$ in the requirement on the data set size n in Theorem 7.2.8. This is possible if the learning algorithm uses only sampling access to labeled examples.

Theorem 7.3.10. *Let $\mathcal{U}$ be a data universe, $\mathcal{Q}$ a set of query descriptions, $\mathcal{GQ}$ a set of distributions over $\mathcal{Q}$, and $P\colon \mathcal{Q}\times\mathcal{U}\to\{0,1\}$ a predicate.*

Then, there is an ε-differentially private (α,β,γ)-accurate data-release algorithm provided that there is an algorithm $\mathcal{L}$ that (γ,β)-learns thresholds over $(\mathcal{Q},\mathcal{GQ}',\{p_u : u\in\mathcal{U}\})$ using $b(n,\gamma,\beta)$ random examples; and we have

$$n \geqslant \frac{C\cdot b(n',\gamma',\beta')\cdot\log\left(\frac{b(n',\gamma',\beta')}{\alpha\gamma\beta}\right)\cdot\log(1/\beta')}{\varepsilon\alpha\gamma},$$

where $n'=\Theta(\log|\mathcal{Q}|/\alpha^2)$, $\beta'=\Theta(\beta\alpha)$, $\gamma'=\Theta(\gamma\alpha)$ and $C>0$ is a sufficiently large constant. If $\mathcal{L}$ runs in time $t(n,\gamma,\beta)$ then the data release algorithm runs in time $\mathrm{poly}(t(n',\gamma',\beta'),n,1/\alpha,\log(1/\beta),1/\gamma)$.

Proof. The proof of this theorem is identical to that of Theorem 7.2.8 except that we put $b_{\text{total}}=2b_{\text{iter}}$ rather than $2kb_{\text{iter}}$. It is easy to check that the algorithm indeed makes only b_{total} *distinct queries* (in the sense of Lemma 7.3.1) to the threshold oracle so that privacy remains ensured. The correctness argument is identical. ■

7.4 First application: data release for conjunctions

With Theorems 7.2.8 and 7.3.10 in hand, we can obtain new data release algorithms "automatically" from learning algorithms that satisfy the properties required by the theorem. In this section we present such data release algorithms for conjunction counting queries using learning algorithms (which require only random examples and work under any distribution) based on polynomial threshold functions.

Throughout this section we fix the query class under consideration to be monotone conjunctions, i.e. we take $\mathcal{U}=\mathcal{Q}=\{0,1\}^d$ and $P(q,u)=1-\bigvee_{i:\,u_i=0}q_i$.

The learning results given later in this section, together with Theorem 7.3.10, immediately yield:

Theorem 7.4.1 (Releasing conjunction counting queries). *1. There is an ε-differentially private algorithm for releasing the class of monotone Boolean conjunction queries over $\mathcal{GQ} = \{$all probability distributions over $\mathcal{Q}\}$ which is (α, β, γ)-accurate and has runtime* $\mathrm{poly}(n)$ *for databases of size n provided that*

$$n \geqslant d^{O\left(d^{1/3}\log\left(\frac{d}{\alpha}\right)^{2/3}\right)} \cdot \widetilde{O}\left(\frac{\log(1/\beta)^3}{\varepsilon\alpha\gamma^2}\right).$$

2. There is an ε-differentially private algorithm for releasing the class of monotone Boolean conjunction queries over $\mathcal{GQ}_w = \{$all probability distributions over $\mathcal{Q}$ supported on $B_w = \{q \in \mathcal{Q} : q_1 + \cdots + q_d \leqslant w\}\}$ which is (α, β, γ)-accurate and has runtime $\mathrm{poly}(n)$ *for databases of size n provided that*

$$n \geqslant d^{O\left(\sqrt{w\log\left(\frac{w\log d}{\alpha}\right)}\right)} \cdot \widetilde{O}\left(\frac{\log(1/\beta)^3}{\varepsilon\alpha\gamma^2}\right).$$

These algorithms are distribution-free, and so we can apply the boosting machinery of [DRV] to get accurate answers to *all* of the w-way conjunctions with similar database size bounds. See the discussion and Corollary 7.1.3 in the introduction.

In Section 7.4.1 we establish structural results showing that certain types of thresholded real-valued functions can be expressed as low-degree polynomial threshold functions. In Section 7.4.2 we state some learning results (for learning under arbitrary distributions) that follow from these representational results. Theorem 7.4.1 above follows immediately from combining the learning results of Section 7.4.2 with Theorem 7.3.10.

7.4.1 Polynomial threshold function representations

Definition 7.4.2. Let $X \subseteq \mathcal{Q} = \{0,1\}^d$ and let f be a Boolean function $f : X \to \{0,1\}$. We say that f has a *polynomial threshold function (PTF) of degree a over X* if there is a real polynomial $A(q_1, \ldots, q_d)$ of degree a such that

$$f(q) = sign(A(q)) \quad \text{for all } q \in X$$

where the *sign* function is $sign(z) = 1$ if $z \geqslant 0$, $sign(z) = 0$ if $z < 0$.

Note that the polynomial A may be assumed without loss of generality to be multilinear since X is a subset of $\{0,1\}^d$.

7.4.1.1 Low-degree PTFs over sparse inputs

Let $B_w \subset \{0,1\}^d$ denote the collection of all points with Hamming weight at most w, i.e. $B_w = \{q \in \{0,1\}^d : q_1 + \cdots + q_d \leqslant w\}$. The main result of this

subsection is a proof that for any $t \in [0,1]$ the function f_t^D has a low-degree polynomial threshold function over B_w.

Lemma 7.4.3. *Fix $t \in [0,1]$. For any database D of size n, the function f_t^D has a polynomial threshold function of degree $O\left(\sqrt{w \log n}\right)$ over the domain B_w.*

To prove Lemma 7.4.3 we will use the following claim:

Claim 7.4.4. *Fix $w > 0$ to be a positive integer and $\varepsilon > 0$. There is a univariate polynomial s of degree $O\left(\sqrt{w \log(1/\varepsilon)}\right)$ which is such that*

1. *$s(w) = 1$; and*
2. *$|s(j)| \leqslant \varepsilon$ for all integers $0 \leqslant j \leqslant w-1$.*

Proof. This claim was proved by Buhrman et al. [BCdWZ], who gave a quantum algorithm which implies the existence of the claimed polynomial (see also Section 1.2 of [She]). Here we give a self-contained construction of a polynomial s with the claimed properties that satisfies the slightly weaker degree bound $\deg(s) = O(\sqrt{w}\log(1/\varepsilon))$. We will use the univariate Chebyshev polynomial C_r of degree $r = \lceil \sqrt{w} \rceil$. Consider the polynomial

$$s(j) = \left(\frac{C_r\left(\frac{j}{w}\left(1+\frac{1}{w}\right)\right)}{C_r\left(1+\frac{1}{w}\right)} \right)^{\lceil \log(1/\varepsilon) \rceil} . \tag{7.14}$$

It is clear that if $j = w$ then $s(j) = 1$ as desired, so suppose that j is an integer $0 \leqslant j \leqslant w-1$. This implies that $(j/w)(1+1/w) < 1$. Now well-known properties of the Chebyshev polynomial (see e.g. [Che]) imply that $|C_r((j/w)(1+1/w))| \leqslant 1$ and $C_r(1+1/w) \geqslant 2$. This gives the $O(\sqrt{w}\log(1/\varepsilon))$ degree bound. ∎

Recall that the predicate function for a data item $u \in \{0,1\}^d$ is denoted by

$$p_u(q) = 1 - \bigvee_{i\colon u_i = 0} q_i .$$

As an easy corollary of Claim 7.4.4 we get:

Corollary 7.4.5. *Fix $\varepsilon > 0$. For every $u \in \{0,1\}^d$, there is a d-variable polynomial A_u of degree $O\left(\sqrt{w \log(1/\varepsilon)}\right)$ which is such that for every $q \in B_w$,*

1. *If $p_u(q) = 1$ then $A_u(q) = 1$;*
2. *If $p_u(q) = 0$ then $|A_u(q)| \leqslant \varepsilon$.*

Proof. Consider the linear function $L(q) = w - \sum_{i:\ u_i=0} q_i$. For $q \in B_w$ we have that $L(q)$ is an integer in $\{0,\ldots,w\}$, and we have $L(q) = w$ if and only if $p_u(q) = 1$. The desired polynomial is $A_u(q) = s(L(q))$. ∎

Proof of Lemma 7.4.3. Consider the polynomial

$$A(q) = \sum_{u \in D} A_u(q)$$

where for each data item u, r_u is the polynomial from Corollary 7.4.5 with its "ε" parameter set to $\varepsilon = 1/(3n)$. We will show that $A(q) - (\lceil tn \rceil - 1/2)$ is the desired polynomial which gives a PTF for f_t^D over B_w.

First, consider any fixed $q \in B_w$ for which $f_t^D(q) = 1$. Such a q must satisfy $f^D(q) = j/n \geqslant t$ for some integer j, and hence $j \geqslant \lceil tn \rceil$. Corollary 7.4.5 now gives that $A(q) \geqslant \lceil tn \rceil - 1/3$.

Next, consider any fixed $q \in B_w$ for which $f_t^D(q) = 0$. Such a q must satisfy $f^D(q) = j/n < t$ for some integer j, and hence $j \leqslant \lceil tn \rceil - 1$. Corollary 7.4.5 now gives that $A(q) \leqslant \lceil tn \rceil - 2/3$. This proves the lemma. ∎

7.4.1.2 Low-degree PTFs over the entire hypercube

Taking $w = d$ in the previous subsection, the results there imply that f_t^D can be represented by a polynomial threshold function of degree $O\left(\sqrt{d \log n}\right)$ over the entire Boolean hypercube $\{0,1\}^d$. In this section we improve the degree to $O\left(d^{1/3}(\log n)^{2/3}\right)$. This result is very similar to Theorem 8 of [KOS] (which is closely based on the main construction and result of [KS]) but with a few differences: first, we use Claim 7.4.4 to obtain slightly improved bounds. Second, we need to use the following notion in place of the notion of the "size of a conjunction" that was used in the earlier results.

Definition 7.4.6. The *width* of a data item $u \in D$ is defined as the number of coordinates i such that $u_i = 0$. The *width* of D is defined as the maximum width of any data item $u \in D$.

We use the following lemma:

Lemma 7.4.7. *Fix any $t \in [0,1]$ and suppose that n-element database D has width w. Then f_t^D has a polynomial threshold function of degree $O\left(\sqrt{w \log n}\right)$ over the domain $\{0,1\}^d$.*

Proof. With the above notion of with the proof is the same as the constructions and arguments of the previous subsection. ∎

Lemma 7.4.8. *Fix any value $r \in \{1,\ldots,d\}$. The function $f_t^D(q_1,\ldots,q_d)$ can be expressed as a decision tree T in which*

1. *each internal node of the tree contains a variable q_i;*
2. *each leaf of T contains a function of the form $f_t^{D'}$ where $D' \subseteq D$ has width at most r;*
3. *the tree T has rank at most $(2d/r)\ln n + 1$.*

Proof sketch. The result follows directly from the proof of Lemma 10 in [KS], except that we use the notion of width from Definition 7.4.6 in place of the notion of the size of a conjunction that is used in [KS]. To see that this works, observe that since $p_u(q) = 1 - \vee_{i:\, u_i=0} q_i$, fixing $q_i = 1$ will fix all predicates p_u with $u_i = 0$ to be zero. Thus the analysis of [KS] goes through unchanged, replacing "terms of f that have size at least r" with "data items in D that have width at least r" throughout. ■

Lemma 7.4.9. *The function f_t^D can be represented as a polynomial threshold function of degree $O(d^{1/3}(\log n)^{2/3})$.*

Proof. The proof is nearly identical to the proof of Theorem 2 in [KS] but with a few small changes. We take r in Lemma 7.4.8 to be $d^{2/3}(\log n)^{1/3}$ and now apply Lemma 7.4.7 to each width-r database D' at a leaf of the resulting decision tree. Arguing precisely as in Theorem 2 of [KS] we get that f_t^D has a polynomial threshold function of degree

$$\max\left\{\tfrac{2d}{r}\ln n + 1, O\left(\sqrt{r\log n}\right)\right\} = O\left(\sqrt{r\log n}\right) = O\left(d^{1/3}(\log n)^{2/3}\right).$$

■

7.4.2 Learning thresholds of conjunction queries under arbitrary distributions

It is well known that using learning algorithms based on polynomial-time linear programming, having low-degree PTFs for a class of functions implies efficient PAC learning algorithms for that class under any distribution using random examples only (see e.g. [KS, HS]). Thus the representational results of Section 7.4.1 immediately give learning results for the class of threshold functions over sums of data items. We state these learning results using the terminology of our reduction below.

Theorem 7.4.10. *Let*

— *$\mathcal{U}$ denote the data universe $\{0,1\}^d$;*

— $\mathcal{Q}$ *denote the set of query descriptions* $\{0,1\}^d$*;*

— $P(q,u) = 1 - \bigvee_{i:\, u_i=0} q_i$ *denote the monotone conjunction predicate;*

— $\mathcal{GQ}$ *denote the set of all probability distributions over* $\mathcal{Q}$*; and*

— $\mathcal{GQ}_w$ *denote the set of all probability distributions over* $\mathcal{Q}$ *that are supported on* $B_w = \{q \in \{0,1\}^d : q_1 + \cdots + q_d \leqslant w\}$.

Then

1. *(Learning thresholds of conjunction queries over all inputs) There is an algorithm* $\mathcal{L}$ *that* (γ,β) *learns thresholds over* $(\mathcal{Q},\mathcal{GQ},\{p_u : u \in \mathscr{U}\})$ *using* $b(n,\gamma,\beta) = d^{O(d^{1/3}(\log n)^{2/3})} \cdot \widetilde{O}(1/\gamma) \cdot \log(1/\beta)$ *queries to an approximate distribution-restricted evaluation oracle for the target n-threshold function (in fact* $\mathcal{L}$ *only uses sampling access to labeled examples). The running time of* $\mathcal{L}$ *is* $\mathrm{poly}(b(n,\gamma,\beta))$.

2. *(Learning thresholds of conjunction queries over sparse inputs) There is an algorithm* $\mathcal{L}$ *that* (γ,β) *learns thresholds over* $(\mathcal{Q},\mathcal{GQ}_w,\{p_u : u \in \mathscr{U}\})$ *using* $b(n,\gamma,\beta) = d^{O((w\log n)^{1/2})} \cdot \widetilde{O}(1/\gamma) \cdot \log(1/\beta)$ *queries to an approximate distribution-restricted evaluation oracle for the target n-threshold function (in fact* $\mathcal{L}$ *only uses sampling access to labeled examples). The running time of* $\mathcal{L}$ *is* $\mathrm{poly}(b(n,\gamma,\beta))$.

Recall from the discussion at the beginning of Section 7.4 that these learning results, together with our reduction, give the private data release results stated at the beginning of the section.

7.5 Second application: data release via Fourier-based learning

In this section we present data release algorithms for parity counting queries and AC^0 counting queries that instantiate our reduction Theorem 7.2.8 with Fourier-based algorithms from the computational learning theory literature. We stress that these algorithms require the more general reduction Theorem 7.2.8 rather than the simpler version of Theorem 7.1.1 because the underlying learning algorithms are not distribution free. We first give our results for parity counting queries in Section 7.5.1 and then our results for AC^0 counting queries in Section 7.5.2.

7.5.1 Parity counting queries using the Harmonic Sieve [Jac]

In this subsection we fix the query class under consideration to be the class of parity queries, i.e. we take $\mathscr{U} = \{0,1\}^d$ and $\mathcal{Q} = \{0,1\}^d$ and we take $P(q,u) = \sum_{i:u_i=1} q_i \pmod 2$ to be the parity predicate. Our main result for releasing parity counting queries is:

Theorem 7.5.1 (Releasing parity counting queries). *There is an ε-differentially private algorithm for releasing the class of parity queries over the uniform distribution on $\mathcal{Q}$ which is (α,β,γ)-accurate and has runtime* $\mathrm{poly}(n)$ *for databases of size n, provided that*

$$n \geqslant \frac{\mathrm{poly}(d, 1/\alpha, 1/\gamma, \log(1/\beta))}{\varepsilon}.$$

This theorem is an immediate consequence of our main reduction, Theorem 7.2.8, and the following learning result:

Theorem 7.5.2. *Let*

- *$\mathscr{U}$ denote the data universe $\{0,1\}^d$;*
- *$\mathcal{Q}$ denote the set of query descriptions $\{0,1\}^d$;*
- *$P(q,u) = \sum_{i:u_i=1} q_i$ (mod 2) denote the parity predicate; and*
- *$\mathcal{GQ}$ contains only the uniform distribution over $\mathcal{Q}$.*

Then there is an algorithm $\mathcal{L}$ that (γ,β) learns thresholds over $(\mathcal{Q},\mathcal{GQ}',\{p_u : u \in \mathscr{U}\})$ where $\mathcal{GQ}'$ is the $(2/\gamma)$-smooth extension of $\mathcal{GQ}$. Algorithm $\mathcal{L}$ uses $b(n,\gamma,\beta) = \mathrm{poly}(d,n,1/\gamma)\cdot\log(1/\beta)$ queries to an approximate G-restricted evaluation oracle for the target n-threshold function when it is learning with respect to a distribution $G \in \mathcal{GQ}'$. The running time of $\mathcal{L}$ is $\mathrm{poly}(b(n,\gamma,\beta))$.

Proof. The claimed algorithm $\mathcal{L}$ is essentially Jackson's Harmonic Sieve algorithm [Jac] for learning Majority of Parities; however, a bit of additional analysis of the algorithm is needed as we now explain.

When Jackson's results on the Harmonic Sieve are expressed in our terminology, they give Theorem 7.5.2 exactly as stated above except for one issue which we now describe. Let G' be any distribution in the $(2/\gamma)$-smooth extension $\mathcal{GQ}'$ of the uniform distribution. In Jackson's analysis, when it is learning a target function f under distribution G', the Harmonic Sieve is given black-box oracle access to f, sampling access to the distribution G', and access to a c-approximation to an evaluation oracle for G', in the following sense: there is some fixed constant $c \in [1/3,3]$ such that when the oracle is queried on $q \in \mathcal{Q}$, it outputs $c \cdot G'[q]$. This is a formally more powerful type of

access to the underlying distribution G' than is allowed in Theorem 7.5.2 since Theorem 7.5.2 only gives $\mathcal{L}$ access to an approximate G'-restricted evaluation oracle for the target function (recall Definition 7.2.5). To be more precise, the only difference is that with the Sieve's black-box oracle access to the target function f it is a priori possible for a learning algorithm to query f even on points where the distribution G' puts zero probability mass, whereas such queries are not allowed for $\mathcal{L}$. Thus to prove Theorem 7.5.2 it suffices to argue that the Harmonic Sieve algorithm, when it is run under distribution G', never needs to make queries on points $q \in \mathcal{Q}$ that have $G'[q] = 0$.

Fortunately, this is an easy consequence of the way the Harmonic Sieve algorithm works. Instead of actually using black-box oracle queries for f, the algorithm actually only ever makes oracle queries to the function $g(q) = 2^d \cdot f(q) \cdot D'[q]$, where D' is a c-approximation to an evaluation oracle for a distribution G'' which is a smooth extension of G'. (See the discussion in Sections 4.1 and 4.2 of [Jac], in particular Steps 16-18 of the `HS` algorithm of Figure 4 and Steps 3 and 5 of the `WDNF` algorithm of Figure 3.) By the definition of a smooth extension, if q is such that $G'[q] = 0$ then $G''[q]$ also equals 0, and consequently $g(q) = 0$ as well. Thus it is straightforward to run the Harmonic Sieve using access to an approximate G'-restricted evaluation oracle: if $G'[q]$ returns 0 then "0" is the correct value of $g(q)$, and otherwise the oracle provides precisely the information that would be available for the Sieve in Jackson's original formulation. ■

7.5.2 AC^0 queries using [JKS]

Fix $\mathcal{U} = \{0,1\}^d$ and $\mathcal{Q} = \{0,1\}^d$. In this subsection we show that our reduction enables us to do efficient private data release for quite a broad class of queries, namely any query computed by a constant-depth circuit.

In more detail, let $P(q,u)\colon \{0,1\}^d \times \{0,1\}^d \to \{0,1\}$ be any predicate that is computed by a circuit of depth $\ell = O(1)$ and size $\mathrm{poly}(d)$. Our data release result for such queries is the following:

Theorem 7.5.3 (Releasing AC^0 queries). *Let $\mathcal{GQ}$ be the set containing the uniform distribution and let $\mathcal{U}, \mathcal{Q}, P$ be as described above. There is an ε-differentially $(\mathcal{U}, \mathcal{Q}, \mathcal{GQ}, P)$ data release algorithm that is (α, β, γ)-accurate and has runtime* $\mathrm{poly}(n)$ *for databases of size n, provided that*

$$n \geqslant d^{O\left(\log^{\ell}\left(\frac{d}{\alpha\gamma}\right)\right)} \cdot \widetilde{O}\left(\frac{\log(1/\beta)^3}{\varepsilon\alpha^2\gamma}\right).$$

See the introduction for a discussion of this result. We observe that given any fixed P as described above, for any given $u \in \mathcal{U} = \{0,1\}^d$ the function

$p_u(q)$ is computed by a circuit of depth ℓ and size poly(d) over the input bits $q_1, \ldots, q_d$. Hence Theorem 7.5.3 is an immediate consequence of Theorem 7.2.8 and the following learning result, which describes the performance guarantee of the quasi-polynomial time algorithm of Jackson et al. [JKS] for learning Majority-of-Parity in our language:

Theorem 7.5.4 (Theorem 9 of [JKS]). *Let*

- *$\mathcal{U}$ denote the data universe $\{0,1\}^d$;*
- *$\mathcal{Q}$ denote the set of query descriptions $\{0,1\}^d$;*
- *$P(q,u)$ be any fixed predicate computed by an AND/OR/NOT circuit of depth $\ell = O(1)$ and size* poly(d)*;*
- *$\mathcal{GQ}$ contains only the uniform distribution over $\mathcal{Q}$; and*
- *$\mathcal{F}$ be the set of all AND/OR/NOT circuits of depth ℓ and size* poly(d).

Then there is an algorithm $\mathcal{L}$ that (γ,β) learns n-thresholds over $(\mathcal{Q},\mathcal{GQ}',\mathcal{F})$ where $\mathcal{GQ}'$ is the $(2/\gamma)$-smooth extension of $\mathcal{GQ}$. Algorithm $\mathcal{L}$ uses approximate distribution restricted oracle access to the function, uses $b(n,\gamma,\beta) = d^{O(\log^\ell(nd/\gamma))} \cdot \log(1/\beta)$ samples and calls to the evaluation oracle, and runs in time $t(n,\gamma,\beta) = d^{O(\log^\ell(nd/\gamma))} \cdot \log(1/\beta)$.

We note that Theorem 9 of [JKS], as stated in that paper, only deals with learning majority-of-AC^0 circuits under the uniform distribution: it says that an n-way Majority of depth-ℓ, size-poly(d) circuits over $\{0,1\}^d$ can be learned to accuracy γ and confidence β under the uniform distribution, using random examples only, in time $d^{O(\log^\ell(nd/\gamma))} \cdot \log(1/\beta)$. However, the boosting-based algorithm of [JKS] is identical in its high-level structure to Jackson's Harmonic Sieve; the only difference is that the [JKS] weak learner simply performs an exhaustive search over all low-weight parity functions to find a weak hypothesis that has non-negligible correlation with the target, whereas the Harmonic Sieve uses a more sophisticated membership-query algorithm (that is an extension of the algorithm of Kushilevitz and Mansour [KM3]). Arguments identical to the ones Jackson gives for the Harmonic Sieve (in Section 7.1 of [Jac]) can be applied unchanged to the [JKS] algorithm, to show that it extends, just like the Harmonic Sieve, to learning under smooth distributions if it is provided with an approximate evaluation oracle for the smooth distribution. In more detail, these arguments show that for any C-smooth distribution G', given sampling access to labeled examples by (G',f) (where f is the target n-way Majority of depth-ℓ, size-poly(d) circuits) and approximate evaluation access to G', the [JKS] algorithm learns f to accuracy

γ and confidence β under G' in time $d^{O(\log^{\ell}(Cnd/\gamma))} \cdot \log(1/\beta)$ This is the result that is restated in our data privacy language above (note that the smoothness parameter there is $C = 2/\gamma$).

7.6 Conclusion and open problems

This work put forward a new reduction from privacy-preserving data analysis to learning thresholds. Instantiating this reduction with various different learning algorithms, we obtained new data release algorithms for a variety of query classes. One notable improvement was for the database size (or error) in distribution-free release of conjunctions and k-way conjunctions. Given these new results, we see no known obstacles for even more dramatic improvements on this central question. In particular, we conclude with the following open question.

Open Question 7.6.1. Is there a differentially private distribution-free data release algorithm (with constant error, e.g., $\alpha = 1/100$) for conjunctions or k-way conjunctions that works for databases of size $\text{poly}(d)$ and runs in time $\text{poly}(n)$ (or $\text{poly}(n, d^k)$ for the case of k-way conjunctions)?

Note that such an algorithm for k-way conjunctions would also imply, via boosting [DRV], that we can privately release *all* k-way conjunctions in time $\text{poly}(n, d^k)$, provided that $|D| \geqslant \text{poly}(d)$.

Chapter 8

Fairness Through Awareness

8.1 Introduction

In this work, we initiate the formal study of fairness in classification. Nearly all classification tasks face the challenge of achieving utility in classification for some purpose, while at the same time preventing discrimination against protected population subgroups. A motivating example is membership in a racial minority in the context of banking. An article in The Wall Street Journal (8/4/2010) describes the practices of CapitalOne.com and its use of the tracking network "[x+1]" to learn detailed demographic information about each visitor to the site, such as approximate income, where she shops, the fact that she rents children's videos, and so on. According to the article, this information is used to "decide which credit cards to show first-time visitors" to the web site, raising the concern of *steering*, namely the (illegal) practice of guiding members of minority groups into less advantageous credit offerings [SA2]. We provide a definition of fairness in classification and a framework for achieving it. Our framework permits us to formulate the question as an optimization problem that can be solved by a linear program. In the remainder of this section we describe our design goals, discuss certain examples that helped to formulate our notion of fairness, and summarize our contributions.

8.1.1 Design Goals

We espouse the following design goals for fair classification:

1. *Arbitrary vendor preferences*. We want to permit the entity that needs to classify individuals, which we call the *vendor*, as much freedom as possible. This allows the vendor to benefit from investment in data mining and market research in designing its classifier, and our system should not need to analyze or vet the vendor's desires. The vendor's wishes should be satisfied subject only to the fairness constraints; and the fairness constraints should be satisfied no matter what the vendor proposes. This absolute guarantee of fairness frees the vendor of any

regulatory concerns: the vendor is assured of compliance with anti-discrimination law because the system cannot discriminate.

2. *Ability to Capture Social Constraints*. Our system should be sufficiently flexible to permit the expression of social constraints such as affirmative action and government regulation.

3. *Prevention of Certain Evils*. Our notion of fairness interdicts a catalogue of discriminatory practices including the following, most of which are described in the next section or in Appendix 8.6.5: redlining; reverse redlining; discrimination based on redundant encodings of membership in the protected set; cutting off business with a segment of the population in which membership in the protected set is disproportionately high; doing business with the "wrong" subset of the protected set (possibly in order to prove a point); and "reverse tokenism."

8.1.2 Potential yet insufficient solutions

We start with some simple and intuitive approaches to achieve these goals, and discuss their limitations:

Fairness through blindness. How might one argue that a personalized website or an advertising system (say, when used for a specific product) does not discriminate based on race? When personalization and advertising decisions are based on months or years of on-line activity, there is a very real possibility that membership in a given demographic group is embedded holographically in the history. Simply deleting, say, the Facebook "sex" and "Interested in men/women" bits almost surely does not hide homosexuality. This point was argued by the (somewhat informal) "Gaydar" study [JM] in which a threshold was found for predicting, based on the sexual preferences of his male friends, whether or not a given male is interested in men. Such *redundant encodings* of sexual preference and other attributes need not be explicitly known or recognized as such, and yet can still have a discriminatory effect.

Fairness at the price of utility. How can a personalization or advertising system, seeking *not* to discriminate, defeat these unknown redundant encodings? Some trivial solutions spring quickly to mind, *e.g.*, classify individuals by means of independent flips of a coin with a fixed bias. Such a solution seems to provide perfect fairness but at a potentially heavy price in utility (specifically, the advertiser prefers to target people who are more likely to buy the product).

Fairness by statistical parity. While the trivial "solutions" are problematic, they do have the nice property of *statistical parity*: the demographics of those receiving positive (or negative) classifications are identical to the demographics of the population as a whole. Although in some cases statistical parity appears to be desirable – in particular, it neutralizes redundant encodings – we now argue its inadequacy as a notion of fairness, presenting three examples in which statistical parity is maintained, but from the point of view of an individual, the outcome is blatantly unfair. In describing these examples, we let S denote the protected set and S^c its complement.

Example 1: Reduced Utility. Consider the following scenario, which we will call Example (1). Suppose in the culture of S the most talented students are steered toward science and engineering and the less talented are steered toward finance, while in the culture of S^c the situation is reversed: the most talented are steered toward finance and those with less talent are steered toward engineering. An organization ignorant of the culture of S and seeking the most talented people may select for "economics," arguably choosing the wrong subset of S, even while maintaining parity. Note that this poor outcome can occur in a "fairness through blindness" approach – the errors come from *ignoring* membership in S.

Example 2: Self-fulfilling prophecy. This is when unqualified members of S are chosen, in order to "justify" future discrimination against S (building a case that there is no point in "wasting" resources on S). Although senseless, this is an example of something pernicious that is not ruled out by statistical parity, showing the weakness of this notion. A variant of this apparently occurs in selecting candidates for interviews: the hiring practices of certain firms are audited to ensure sufficiently many *interviews* of minority candidates, but less care is taken to ensure that the best minorities – those that might actually compete well with the better non-minority candidates – are invited [Zar].

Example 3: Subset Targeting. Statistical parity for S does not imply statistical parity for subsets of S. This can be maliciously exploited in many ways. For example, consider an advertisement for a product X which is targeted to members of S that are likely to be interested in X and to members of S^c that are very unlikely to be interested in X. *Clicking* on such an ad may be strongly correlated with membership in S (even if exposure to the ad obeys statistical parity).

8.1.3 Our contributions

Our main contribution lies in formulating and conceptualizing the problem. We now summarize the key features of our framework.

Fairness is individual-based rather than group-based, and fundamentally requires an understanding of the similarities and differences among the individuals involved. We capture fairness by the principle that people who are similar with *with respect to a particular task* should be classified similarly. This gives *fairness for individuals*, while statistical parity speaks to *fairness for a group*. In order to accomplish individual-based fairness, we assume a distance metric that defines the similarity between the individuals. This is the source of "awareness" in the title of this chapter. In example (1) above, an appropriate distance function would indicate that science students in S are more similar to the finance students in S^c than to science students in S^c. We believe that reliance on this sort of knowledge is unavoidable, given that the presence of redundant encodings rules out "fairness through blindness." Our approach is also consistent with many real-life discrimination cases in which the complaint is individual-based, such as the recent class-action lawsuit against Wal-Mart begun when a female employee sued based on the fact that a male with the same job title and less experience was paid $10,000 a year more than she.

Sunlight for the Metric. Justifying the availability of a distance metric in various settings is one of the most challenging aspects of our framework. It is our contention that metrics are employed implicitly or explicitly in many classification settings, such as admissions procedures, advertising ("people who buy X and live in zip code Y are similar to people who live in zip code Z and buy W"), and loan applications (credit scores). One contribution of this work is making the distance metric explicit, and open to discussion, debate, and revision. Indeed, we envision that, typically, the distance metric would be externally imposed, for example, by a regulatory body, or externally proposed, by a civil rights organization. We discuss the acquisition of the distance metric and related open problems in Section 8.6.

Formulation of the problem as an optimization problem, which can be expressed as a linear program. Our framework boils down to finding a mapping from individuals to distributions over outcomes that minimizes expected loss, subject to a *Lipschitz condition* requiring that similar people (as defined

by the distance metric) are mapped similarly. (If statistical parity is desired, this condition can be added to the optimization problem.)

Close relationship to privacy. We observe that our definition of fairness is a generalization of the notion of differential privacy [Dwo1, DMNS]. We draw an analogy between individuals in the setting of fairness and databases in the setting of differential privacy. We exploit this analogy to obtain a more efficient instantiation of our general approach. We also touch on the extent to which fairness can provide privacy, in the context of online advertising.

Having laid out the framework we present the following results:

— In Section 8.3, we give conditions on the similarity metric, via an earthmover distance, such that fairness for individuals (the Lipschitz condition) yields group fairness (statistical parity). More precisely, we show that the Lipschitz condition implies statistical parity between two groups if and only if the Earthmover distance between the two groups is small. This characterization is an important tool in understanding the consequences of imposing the Lipschitz condition.

— In Section 8.4, we give techniques for forcing statistical parity when it is not implied by the Lipschitz condition (the case of preferential treatment), while preserving as much fairness for individuals as possible.

— In Section 8.5, we exploit the relationship with differential privacy to develop a more efficient variation of our fairness mechanism. We prove that our solution has small error when the metric space of individuals has small doubling dimension, a natural condition arising in machine learning applications. We also prove a lower bound showing that *any* mapping satisfying the Lipschitz condition has error that scales with the doubling dimension.

We are unaware of other work on the topic of fairness in classification. Two related topics are learning and fair division (cake-cutting). We postpone our discussion of these topics and other related work until after we have developed our framework.

The remainder of the chapter is organized as follows. In Section 8.2 we expand on our formulation of the problem, providing notation and definitions of its key components. Section 8.2.1 presents the linear program, and an example of how it achieves fairness. In the following three sections, we present the three results described above. Finally, in the Discussion we consider significant open questions and directions.

8.2 Formulation of the problem

In this section we describe our setup in its most basic form. We shall later see generalizations of this basic formulation. *Individuals* are the objects to be classified; we denote the set of individuals by V. In this chapter we consider classifiers that map individuals to outcomes. We denote the set of outcomes by A. In the simplest non-trivial case $A = \{0, 1\}$. To ensure fairness, we will consider randomized classifiers mapping individuals to distributions over outcomes. To introduce our notion of fairness we assume the existence of a metric on individuals $d\colon V \times V \to \mathbb{R}$. We will consider randomized mappings $M\colon V \to \mu(A)$ from individuals to probability distributions over outcomes. Such a mapping naturally describes a randomized classification procedure: to classify $x \in V$ choose an outcome a according to the distribution $M(x)$. We interpret the goal of "mapping similar people similarly" to mean that the distributions assigned to similar people are similar. Later we will discuss two specific measures of similarity of distributions, D_∞ and D_{tv}, of interest in this work.

Definition 8.2.1 (Lipschitz mapping)**.** A mapping $M\colon V \to \mu(A)$ satisfies the (D, d)-*Lipschitz* property if for every $x, y \in V$, we have

$$D(Mx, My) \leqslant d(x, y). \tag{8.1}$$

We note that there always exists a Lipschitz classifier, for example, by mapping all individuals to the same distribution over A. Which classifier we shall choose thus depends on a notion of utility. We capture utility using a *loss function* $L\colon V \times A \to \mathbb{R}$. This setup naturally leads to the optimization problem:

> Find a mapping from individuals to distributions over outcomes that minimizes expected loss subject to the Lipschitz condition.

8.2.1 Achieving Fairness

Our fairness definition leads to an optimization problem in which we minimize an arbitrary loss function $L\colon V \times A \to \mathbb{R}$ while achieving the d-Lipschitz property for a given metric $d\colon V \times V \to \mathbb{R}$. We denote by $\mathcal{I}$ an instance of our problem consisting of a metric $d\colon V \times V \to \mathbb{R}$, and a loss function $L\colon V \times A \to \mathbb{R}$. We denote the optimal value of the minimization problem by $\text{opt}(\mathcal{I})$, as formally defined in Figure 8.1. We will also write the mapping $M\colon V \to \mu(A)$ as $M = \{\mu_x\}_{x \in V}$ where $\mu_x = M(x) \in \mu(A)$.

$$\mathrm{opt}(\mathcal{I}) \stackrel{\text{def}}{=} \min_{\{\mu_x\}_{x\in V}} \quad \mathop{\mathbb{E}}_{x\sim V} \mathop{\mathbb{E}}_{a\sim\mu_x} L(x,a) \tag{8.2}$$

$$\text{subject to} \quad \forall x,y \in V, : \quad D(\mu_x,\mu_y) \leqslant d(x,y) \tag{8.3}$$

$$\forall x \in V : \quad \mu_x \in \mu(A) \tag{8.4}$$

Figure 8.1: The Fairness LP: Loss minimization subject to fairness constraint

Probability Metrics The first choice for D that may come to mind is the statistical distance: Let P, Q denote probability measures on a finite domain A. The *statistical distance* or *total variation norm* between P and Q is denoted by

$$D_{\mathrm{tv}}(P,Q) = \frac{1}{2}\sum_{a\in A} |P(a) - Q(a)|. \tag{8.5}$$

The following lemma is easily derived from the definitions of $\mathrm{opt}(\mathcal{I})$ and D_{tv}.

Lemma 8.2.2. *Let $D = D_{\mathrm{tv}}$. Given an instance $\mathcal{I}$ we can compute* $\mathrm{opt}(\mathcal{I})$ *with a linear program of size* $\mathrm{poly}(|V|,|A|)$.

Remark 8.2.3. When dealing with the set V, we have assumed that V is the set of real individuals (rather than the potentially huge set of all possible encodings of individuals). More generally, we may only have access to a subsample from the set of interest. In such a case, there is the additional challenge of extrapolating a classifier over the entire set.

A weakness of using D_{tv} as the distance measure on distributions, is that we should then assume that the distance metric (measuring distance between individuals) is scaled such that for similar individuals $d(x,y)$ is very close to zero, while for very dissimilar individuals $d(x,y)$ is close to one. A potentially better choice for D in this respect is sometimes called *relative ℓ_∞ metric*:

$$D_\infty(P,Q) = \sup_{a\in A} \log\left(\max\left\{\frac{P(a)}{Q(a)}, \frac{Q(a)}{P(a)}\right\}\right). \tag{8.6}$$

With this choice we think of two individuals x, y as similar if $d(x,y) \ll 1$. In this case, the Lipschitz condition in Equation 8.1 ensures that x and y map to similar distributions over A. On the other hand, when x, y are very dissimilar, i.e., $d(x,y) \gg 1$, the condition imposes only a weak constraint on the two corresponding distributions over outcomes.

Lemma 8.2.4. *Let $D = D_\infty$. Given an instance $\mathcal{I}$ we can compute* $\mathrm{opt}(\mathcal{I})$ *with a linear program of size* $\mathrm{poly}(|V|,|A|)$.

Proof. We note that the objective function and the first constraint are indeed linear in the variables $\mu_x(a)$, as the first constraint boils down to requirements of the form $\mu_x(a) \leqslant e^{d(x,y)}\mu_y(a)$. The second constraint $\mu_x \in \mu(A)$ can easily be rewritten as a set of linear constraints. ∎

Notation. Recall that we often write the mapping $M\colon V \to \mu(A)$ as $M = \{\mu_x\}_{x\in V}$ where $\mu_x = M(x) \in \mu(A)$. In this case, when S is a distribution over V we denote by μ_S the distribution over A defined as $\mu_S(a) = \mathbb{E}_{x\sim S}\,\mu_x(a)$ where $a \in A$.

Useful Facts It is not hard to check that both D_{tv} and D_∞ are metrics with the following properties.

Lemma 8.2.5. $D_{\mathrm{tv}}(P,Q) \leqslant 1 - \exp(-D_\infty(P,Q)) \leqslant D_\infty(P,Q)$

Fact 8.2.6. *For any three distributions P, Q, R and non-negative numbers $\alpha, \beta \geqslant 0$ such that $\alpha + \beta = 1$, we have $D_{\mathrm{tv}}(\alpha P + \beta Q, R) \leqslant \alpha D_{\mathrm{tv}}(P,R) + \beta D_{\mathrm{tv}}(Q,R)$.*

Post-Processing. An important feature of our definition is that it behaves well with respect to *post-processing*. Specifically, if $M\colon V \to \mu(A)$ is (D,d)-Lipschitz for $D \in \{D_{\mathrm{tv}}, D_\infty\}$ and $f : A \to B$ is any possibly randomized function from A to another set B, then the composition $f \circ M\colon V \to \mu(B)$ is a (D,d)-Lipschitz mapping. This would in particular be useful in the setting of the example in Section 8.2.2.

8.2.2 Example: Ad network

Here we expand on the example of an advertising network mentioned in the Introduction. We explain how the Fairness LP provides a fair solution protecting against the evils described in Appendix 8.6.5. The Wall Street Journal article [SA2] describes how the [x+1] tracking network collects demographic information about individuals, such as their browsing history, geographical location, and shopping behavior, and utilizes this to assign a person to one of 66 groups. For example, one of these groups is "White Picket Fences," a market segment with median household income of just over $50,000, aged 25 to 44 with kids, with some college education, etc. Based on this assignment to a group, CapitalOne decides which credit card, with particular terms of credit, to show the individual. In general we view a classification task as involving two distinct parties: the *data owner* is a trusted party holding the data of individuals, and the *vendor* is the party that wishes to classify individuals. The loss function may be defined solely by either party or by both parties in

collaboration. In this example, the data owner is the ad network [x+1], and the vendor is CapitalOne.

The ad network ([x+1]) maintains a mapping from individuals into categories. We can think of these categories as outcomes, as they determine which ads will be shown to an individual. In order to comply with our fairness requirement, the mapping from individuals into categories (or outcomes) will have to be randomized and satisfy the Lipschitz property introduced above. Subject to the Lipschitz constraint, the ad network can still express its own belief as to how individuals should be assigned to categories using the loss function. However, since the Lipschitz condition is a hard constraint there is no possibility of discriminating between individuals that are deemed similar by the metric. In particular, this will disallow arbitrary distinctions between protected individuals, thus preventing both reverse tokenism and the self-fulfilling prophecy. In addition, the metric can eliminate the existence of redundant encodings of certain attributes thus also preventing redlining of those attributes. In Section 8.3 we will see a characterization of which attributes are protected by the metric in this way.

8.2.3 Connection to Differential Privacy

Our notion of fairness may be viewed as a generalization of differential privacy [Dwo1, DMNS]. As it turns out our notion can be seen as a generalization of differential privacy. To see this, consider a simple setting of differential privacy where a *database curator* maintains a database x (thought of as a subset of some universe U) and a data analyst is allowed to ask a query $F\colon V \to A$ on the database. Here we denote the set of databases by $V = 2^U$ and the range of the query by A. A mapping $M\colon V \to \mu(A)$ satisfies *ε-differential privacy* if and only if M satisfies the (D_∞, d)-Lipschitz property, where, letting $x \triangle y$ denote the symmetric difference between x and y, we define $d(x,y) \stackrel{\text{def}}{=} \varepsilon|x \triangle y|$.

The utility loss of the analyst for getting an answer $a \in A$ from the mechanism is defined as $L(x,a) = d_A(Fx,a)$, that is distance of the true answer from the given answer. Here distance refers to some distance measure in A that we described using the notation d_A. For example, when $A = \mathbb{R}$, this could simply be $d_A(a,b) = |a-b|$. The optimization problem (8.2) in Figure 8.1 (*i.e.*, $\text{opt}(\mathcal{I}) \stackrel{\text{def}}{=} \min \mathbb{E}_{x\sim V}\, \mathbb{E}_{a\sim\mu_x} L(x,a)$) now defines the optimal differentially private mechanism in this setting. We can draw a conceptual analogy between the utility model in differential privacy and that in fairness. If we think of outcomes as representing information about an individual, then the vendor wishes to receive what she believes is the most "accurate" representation of an individual. This is quite similar to the goal of the analyst in differential privacy.

In the current work we deal with more general metric spaces than in differential privacy. Nevertheless, we later see (specifically in Section 8.5) that some of the techniques used in differential privacy carry over to the fairness setting.

8.3 Relationship between Lipschitz property and statistical parity

In this section we discuss the relationship between the Lipschitz property articulated in Definition 8.2.1 and *statistical parity*. As we discussed earlier, statistical parity is insufficient as a general notion of fairness. Nevertheless statistical parity can have several desirable features, *e.g.*, as described in Proposition 8.3.2 below. In this section we demonstrate that the Lipschitz condition naturally implies statistical parity between certain subsets of the population.

Formally, statistical parity is the following property.

Definition 8.3.1 (Statistical parity). We say that a mapping $M\colon V \to \mu(A)$ satisfies *statistical parity* between distributions S and T up to bias ε if

$$D_{\mathrm{tv}}(\mu_S, \mu_T) \leqslant \varepsilon\,. \tag{8.7}$$

Proposition 8.3.2. *Let $M\colon V \to \mu(A)$ be a mapping that satisfies statistical parity between two sets S and T up to bias ε. Then, for every set of outcomes $O \subseteq A$, we have the following two properties.*

1. $|\mathbb{P}\{M(x) \in O \mid x \in S\} - \mathbb{P}\{M(x) \in O \mid x \in T\}| \leqslant \varepsilon$,
2. $|\mathbb{P}\{x \in S \mid M(x) \in O\} - \mathbb{P}\{x \in T \mid M(x) \in O\})| \leqslant \varepsilon\,.$

Intuitively, this proposition says that if M satisfies statistical parity, then members of S are equally likely to observe a set of outcomes as are members of T. Furthermore, the fact that an individual observed a particular outcome provides no information as to whether the individual is a member of S or a member of T. We can always choose $T = S^c$ in which case we compare S to the general population.

A fundamental question that arises in our approach is: *When does the Lipschitz condition imply statistical parity between two distributions S and T on V?* We will see that the answer to this question is closely related to the *earthmover distance* between S and T, which we will define shortly.

The next definition formally introduces the quantity that we will study, that is, the extent to which any Lipschitz mapping can violate statistical parity. In other words, we answer the question, "How biased with respect to S and T might the solution of the fairness LP be, in the worst case?"

Definition 8.3.3 (Bias). We define

$$\mathrm{bias}_{D,d}(S,T) \stackrel{\mathrm{def}}{=} \max \mu_S(0) - \mu_T(0), \tag{8.8}$$

where the maximum is taken over all (D,d)-Lipschitz mappings $M = \{\mu_x\}_{x\in V}$ mapping V into $\mu(\{0,1\})$.

Note that $\mathrm{bias}_{D,d}(S,T) \in [0,1]$. Even though in the definition we restricted ourselves to mappings into distributions over $\{0,1\}$, it turns out that this is without loss of generality, as we show next.

Lemma 8.3.4. *Let $D \in \{D_{\mathrm{tv}}, D_\infty\}$ and let $M\colon V \to \mu(A)$ be any (D,d)-Lipschitz mapping. Then, M satisfies statistical parity between S and T up to* $\mathrm{bias}_{D,d}(S,T)$.

Proof. Let $M = \{\mu_x\}_{x\in V}$ be any (D,d)-Lipschitz mapping into A. We will construct a (D,d)-Lipschitz mapping $M'\colon V \to \mu(\{0,1\})$ which has the same bias between S and T as M.

Indeed, let $A_S = \{a \in A\colon \mu_S(a) > \mu_T(a)\}$ and let $A_T = A_S^c$. Put $\mu'_x(0) = \mu_x(A_S)$ and $\mu'_x(1) = \mu_x(A_T)$. We claim that $M' = \{\mu'_x\}_{x\in V}$ is a (D,d)-Lipschitz mapping. In both cases $D \in \{D_{\mathrm{tv}}, D_\infty\}$ this follows directly from the definition. On the other hand, it is easy to see that

$$D_{\mathrm{tv}}(\mu_S,\mu_T) = D_{\mathrm{tv}}(\mu'_S,\mu'_T) = \mu'_S(0) - \mu'_T(0) \leqslant \mathrm{bias}_{D,d}(S,T). \qquad \blacksquare$$

Earthmover Distance. We will presently relate $\mathrm{bias}_{D,d}(S,T)$ for $D \in \{D_{\mathrm{tv}}, D_\infty\}$ to certain *earthmover distances* between S and T, which we define next.

Definition 8.3.5 (Earthmover distance). Let $\sigma\colon V \times V \to \mathbb{R}$ be a nonnegative distance function. The σ-earthmover distance between two distributions S and T, denoted $\sigma_{\mathrm{EM}}(S,T)$, is defined as the value of the so-called *Earthmover LP*:

$$\begin{aligned} \sigma_{\mathrm{EM}}(S,T) \stackrel{\mathrm{def}}{=} \quad \min \quad & \sum_{x,y\in V} h(x,y)\sigma(x,y) \\ \text{subject to} \quad & \sum_{y\in V} h(x,y) = S(x) \\ & \sum_{y\in V} h(y,x) = T(x) \\ & h(x,y) \geqslant 0 \end{aligned}$$

We will need the following standard lemma, which simplifies the definition of the earthmover distance in the case where σ is a metric.

Lemma 8.3.6. *Let $d\colon V \times V \to \mathbb{R}$ be a metric. Then,*

$$\begin{aligned} d_{\mathrm{EM}}(S,T) = \quad & \min \quad \sum_{x,y\in V} h(x,y)d(x,y) \\ & \text{subject to} \quad \sum_{y\in V} h(x,y) = \sum_{y\in V} h(y,x) + S(x) - T(x) \\ & \qquad\qquad h(x,y) \geqslant 0 \end{aligned}$$

Theorem 8.3.7. *Let d be a metric. Then,*

$$\mathrm{bias}_{D_{\mathrm{tv}},d}(S,T) \leqslant d_{\mathrm{EM}}(S,T). \tag{8.9}$$

If furthermore $d(x,y) \leqslant 1$ for all x, y, then we have

$$\mathrm{bias}_{D_{\mathrm{tv}},d}(S,T) \geqslant d_{\mathrm{EM}}(S,T). \tag{8.10}$$

Proof. The proof is by linear programming duality. We can express $\mathrm{bias}_{D_{\mathrm{tv}},d}(S,T)$ as the following linear program:

$$\begin{aligned} \mathrm{bias}(S,T) = \quad & \max \quad \sum_{x\in V} S(x)\mu_x(0) - \sum_{x\in V} T(x)\mu_x(0) \\ & \text{subject to} \quad \mu_x(0) - \mu_y(0) \leqslant d(x,y) \\ & \qquad\qquad \mu_x(0) + \mu_x(1) = 1 \\ & \qquad\qquad \mu_x(a) \geqslant 0 \end{aligned}$$

Here, we used the fact that

$$D_{\mathrm{tv}}(\mu_x,\mu_y) \leqslant d(x,y) \quad \Longleftrightarrow \quad \left|\mu_x(0) - \mu_y(0)\right| \leqslant d(x,y).$$

The constraint on the RHS is enforced in the linear program above by the two constraints $\mu_x(0) - \mu_y(0) \leqslant d(x,y)$ and $\mu_y(0) - \mu_x(0) \leqslant d(x,y)$.

We can now prove (8.9). Since d is a metric, we can apply Lemma 8.3.6. Let $\{f(x,y)\}_{x,y\in V}$ be a solution to the LP defined in Lemma 8.3.6. By putting $\varepsilon_x = 0$ for all $x \in V$, we can extend this to a feasible solution to the LP defining $\mathrm{bias}(S,T)$ achieving the same objective value. Hence, we have $\mathrm{bias}(S,T) \leqslant d_{\mathrm{EM}}(S,T)$.

Let us now prove (8.10), using the assumption that $d(x,y) \leqslant 1$. To do so, consider dropping the constraint that $\mu_x(0) + \mu_x(1) = 1$ and denote by $\beta(S,T)$ the resulting LP:

$$\begin{aligned} \beta(S,T) \stackrel{\mathrm{def}}{=} \quad & \max \quad \sum_{x\in V} S(x)\mu_x(0) - \sum_{x\in V} T(x)\mu_x(0) \\ & \text{subject to} \quad \mu_x(0) - \mu_y(0) \leqslant d(x,y) \\ & \qquad\qquad \mu_x(0) \geqslant 0 \end{aligned}$$

It is clear that $\beta(S,T) \geqslant \mathrm{bias}(S,T)$ and we claim that in fact $\mathrm{bias}(S,T) \geqslant \beta(S,T)$. To see this, consider any solution $\{\mu_x(0)\}_{x\in V}$ to $\beta(S,T)$. Without changing the objective value we may assume that $\min_{x\in V} \mu_x(0) = 0$. By our assumption that $d(x,y) \leqslant 1$ this means that $\max_{x\in V} \mu_x(0) \leqslant 1$. Now put $\mu_x(1) = 1 - \mu_x(0) \in [0,1]$. This gives a solution to $\mathrm{bias}(S,T)$ achieving the same objective value. We therefore have,

$$\mathrm{bias}(S,T) = \beta(S,T).$$

On the other hand, by strong LP duality, we have

$$\begin{aligned} \beta(S,T) = \quad & \min \quad \sum_{x,y\in V} h(x,y)d(x,y) \\ & \text{subject to} \quad \sum_{y\in V} h(x,y) \geqslant \sum_{y\in V} h(y,x) + S(x) - T(x) \\ & \qquad\qquad h(x,y) \geqslant 0 \end{aligned}$$

It is clear that in the first constraint we must have equality in any optimal solution. Otherwise we can improve the objective value by decreasing some variable $h(x,y)$ without violating any constraints.

Since d is a metric we can now apply Lemma 8.3.6 to conclude that $\beta(S,T) = d_{\mathrm{EM}}(S,T)$ and thus $\mathrm{bias}(S,T) = d_{\mathrm{EM}}(S,T)$. ∎

Remark 8.3.8. Here we point out a different proof of the fact that $\mathrm{bias}_{D_{\mathrm{tv}},d}(S,T) \leqslant d_{\mathrm{EM}}(S,T)$ which does not involve LP duality. Indeed $d_{\mathrm{EM}}(S,T)$ can be interpreted as giving the cost of the best *coupling* between the two distributions S and T subject to the penalty function $d(x,y)$. Recall, a coupling is a distribution (X,Y) over $V \times V$ such that the marginal distributions are S and T, respectively. The cost of the coupling is $\mathbb{E}\, d(X,Y)$. It is not difficult to argue directly that any such coupling gives an upper bound on $\mathrm{bias}_{D_{\mathrm{tv}},d}(S,T)$. We chose the linear programming proof since it leads to additional insight into the tightness of the theorem.

The situation for $\mathrm{bias}_{D_\infty,d}$ is somewhat more complicated and we do not get a tight characterization in terms of an earthmover distance. We do however have the following upper bound.

Lemma 8.3.9.

$$\mathrm{bias}_{D_\infty,d}(S,T) \leqslant \mathrm{bias}_{D_{\mathrm{tv}},d}(S,T) \tag{8.11}$$

Proof. By Lemma 8.2.5, we have $D_{\mathrm{tv}}(\mu_x,\mu_y) \leqslant D_\infty(\mu_x,\mu_y)$ for any two distributions μ_x,μ_y. Hence, every (D_∞, d)-Lipschitz mapping is also (D_{tv}, d)-Lipschitz. Therefore, $\mathrm{bias}_{D_{\mathrm{tv}},d}(S,T)$ is a relaxation of $\mathrm{bias}_{D_\infty,d}(S,T)$. ∎

Corollary 8.3.10.

$$\text{bias}_{D_\infty,d}(S,T) \leqslant d_{\text{EM}}(S,T) \tag{8.12}$$

For completeness we note the dual linear program obtained from the definition of $\text{bias}_{D_\infty,d}(S,T)$:

$$\begin{aligned}
\text{bias}_{D_\infty,d}(S,T) = \quad \min \quad & \sum_{x\in V} \varepsilon_x \\
\text{subject to} \quad & \sum_{y\in V} f(x,y) + \varepsilon_x \geqslant \sum_{y\in V} f(y,x)e^{d(x,y)} + S(x) - T(x) \qquad (8.13) \\
& \sum_{y\in V} g(x,y) + \varepsilon_x \geqslant \sum_{y\in V} g(y,x)e^{d(x,y)} \qquad (8.14) \\
& f(x,y), g(x,y) \geqslant 0
\end{aligned}$$

Similar to the proof of Theorem 8.3.7, we may interpret this program as a *flow* problem. The variables $f(x,y), g(x,y)$ represent a nonnegative *flow* from x to y and ε_x are slack variables. Note that the variables ε_x are unrestricted as they correspond to an equality constraint. The first constraint requires that x has at least $S(x) - T(x)$ outgoing units of flow in f. The RHS of the constraints states that the penalty for *receiving* a unit of flow from y is $e^{d(x,y)}$. However, it is no longer clear that we can get rid of the variables $\varepsilon_x, g(x,y)$.

Open Question 8.3.1. Can we achieve a tight characterization of when (D_∞, d)-Lipschitz implies statistical parity?

8.4 Preferential Treatment

In this section, we explore how to implement what may be called *fair preferential treatment*. Indeed, a typical question when we discuss fairness is, *"What if we want to ensure statistical parity between two groups S and T, but members of S are less likely to be "qualified"?*

For example, in the context of bank loans, it may be the case that the members of S are generally less well off financially than members of T, in which case, the argument goes, statistical parity might be inappropriate. In Section 8.3, we have seen that when S and T are "similar" then the Lipschitz condition implies statistical parity. Here we consider the complementary case where S and T are very different and imposing statistical parity corresponds to preferential treatment. This is a cardinal question, which we examine with a concrete example illustrated in Figure 8.2.

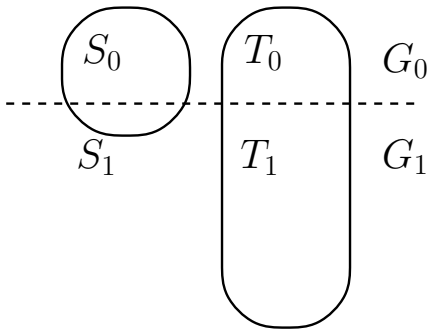

Figure 8.2: $S_0 = G_0 \cap S$, $T_0 = G_0 \cap T$

For simplicity, let $T = S^c$. Assume $|S|/|T \cup S| = 1/10$, so S is only 10% of the population. Suppose that our task-specific metric partitions $S \cup T$ into two groups, call them G_0 and G_1, where members of G_i are very close to one another and very far from all members of G_{1-i}. Let S_i, respectively T_i, denote the intersection $S \cap G_i$, respectively $T \cap G_i$, for $i = 0, 1$. Finally, assume $|S_0| = |T_0| = 9|S|/10$. Thus, G_0 contains less than 20% of the total population, and is equally divided between S and T.

The Lipschitz condition requires that members of each G_i be treated similarly to one another, but there is no requirement that members of G_0 be treated similarly to members of G_1. The treatment of members of S, on average, may therefore be very different from the treatment, on average, of members of T, since members of S are over-represented in G_0 and under-represented in G_1. Thus the Lipschitz condition says nothing about statistical parity in this case.

Suppose the members of G_i are to be shown an advertisement ad_i for a loan offering, where the terms in ad_1 are superior to those in ad_0. Suppose further that the distance metric has partitioned the population according to (something correlated with) credit score, with those in G_1 having higher scores than those in G_0.

On the one hand, this seems fair: people with better ability to repay are being shown a more attractive product. Now we ask two questions: "What is the effect of imposing statistical parity?" and "What is the effect of failing to impose statistical parity?"

Imposing Statistical Parity. Essentially all of S is in G_0, so for simplicity let us suppose that indeed $S_0 = S \subset G_0$. In this case, to ensure that members of S have comparable chance of seeing ad_1 as do members of T, members of S must be treated, for the most part, like those in T_1. In addition, by the Lipschitz condition, members of T_0 must be treated like members of $S_0 = S$, so these, also, are treated like T_1, and the space essentially collapses, leaving

only trivial solutions such as assigning a fixed probability distribution on the advertisements $(\mathrm{ad}_0, \mathrm{ad}_1)$ and showing ads according to this distribution to each individual, or showing all individuals ad_i for some fixed i. However, while fair (all individuals are treated identically), these solutions fail to take the vendor's loss function into account.

Failing to Impose Statistical Parity. The demographics of the groups G_i differ from the demographics of the general population. Even though half the individuals shown ad_0 are members of S and half are members of T, this in turn can cause a problem with fairness: an "anti-S" vendor can effectively eliminate most members of S by replacing the "reasonable" advertisement ad_0 offering less good terms, with a blatantly hostile message designed to drive away customers. This eliminates essentially all business with members of S, while keeping intact most business with members of T. Thus, if members of S are relatively far from the members of T according to the distance metric, then satisfying the Lipschitz condition may fail to prevent some of the unfair practices.

8.4.1 An alternative optimization problem

With the above discussion in mind, we now suggest a different approach, in which we insist on statistical parity, but we relax the Lipschitz condition between elements of S and elements of S^c. This is consistent with the essence of preferential treatment, which implies that elements in S are treated differently than elements in T. The approach is inspired by the use of the earthmover relaxation in the context of metric labeling and 0-extension [CKNZ, Cha]. Relaxing the $S \times T$ Lipschitz constraints also makes sense if the information about the distances between members of S and members of T is of lower quality, or less reliable, than the internal distance information within these two sets.

We proceed in two steps:

1. (a) First we compute a mapping from elements in S to distributions over T which transports the uniform distribution over S to the uniform distribution over T, while minimizing the total distance traveled. Additionally the mapping preserves the Lipschitz condition between elements within S.

   (b) This mapping gives us the following new loss function for elements of T: For $y \in T$ and $a \in A$ we define a new loss, $L'(y, a)$, as

$$L'(y, a) = \sum_{x \in S} \mu_x(y) L(x, a) + L(y, a),$$

where $\{\mu_x\}_{x\in S}$ denotes the mapping computed in step (a). L' can be viewed as a reweighting of the loss function L, taking into account the loss on S (indirectly through its mapping to T).

2. Run the Fairness LP only on T, using the new loss function L'.

Composing these two steps yields a a mapping from $V = S \cup T$ into A.

Formally, we can express the first step of this alternative approach as a restricted Earthmover problem defined as

$$\begin{aligned} d_{\text{EM+L}}(S,T) \stackrel{\text{def}}{=} \quad \min \quad & \mathop{\mathbb{E}}_{x\in S} \mathop{\mathbb{E}}_{y\sim\mu_x} d(x,y) \\ \text{subject to} \quad & D(\mu_x,\mu_{x'}) \leqslant d(x,x') \quad \text{for all} \quad x,x' \in S \\ & D_{\text{tv}}(\mu_S, U_T) \leqslant \varepsilon \\ & \mu_x \in \mu(T) \quad \text{for all} \quad x \in S \end{aligned} \tag{8.15}$$

Here, U_T denotes the uniform distribution over T. Given $\{\mu_x\}_{x\in S}$ which minimizes (8.15) and $\{\nu_x\}_{x\in T}$ which minimizes the original fairness LP (8.2) restricted to T, we define the mapping $M\colon V \to \mu(A)$ by putting

$$M(x) = \begin{cases} \nu_x & x \in T \\ \mathbb{E}_{y\sim\mu_x} \nu_y & x \in S \end{cases}. \tag{8.16}$$

Before stating properties of the mapping M we make some remarks.

1. Fundamentally, this new approach shifts from minimizing loss, subject to the parity and Lipschitz constraints, to minimizing loss and disruption of $S \times T$ Lipschitz requirement, subject to the parity and $S \times S$ and $T \times T$ Lipschitz constraints. This gives us a bicriteria optimization problem, with a wide range of options.

2. We also have some flexibility even in the current version. For example, we can eliminate the re-weighting, prohibiting the vendor from expressing any opinion about the fate of elements in S. This makes sense in several settings. For example, the vendor may *request* this due to ignorance (*e.g.*, lack of market research) about S, or the vendor may have some (hypothetical) special legal status based on past discrimination against S.

3. It is instructive to compare the alternative approach to a modification of the Fairness LP in which we enforce statistical parity and eliminate the Lipschitz requirement on $S \times T$. The alternative approach is more

faithful to the $S \times T$ distances, providing protection against the self-fulfilling prophecy discussed in the Introduction, in which the vendor deliberately selects the "wrong" subset of S while still maintaining statistical parity.

4. A related approach to addressing preferential treatment involves *adjusting* the metric in such a way that the Lipschitz condition will imply statistical parity. This coincides with at least one philosophy behind affirmative action: that the metric does not fully reflect potential that may be undeveloped because of unequal access to resources. Therefore, when we consider one of the strongest individuals in S, affirmative action suggests it is more appropriate to consider this individual as similar to one of the strongest individuals of T (rather than to an individual of T which is close according to the original distance metric). In this case, it is natural to adjust the distances between elements in S and T rather than inside each one of the populations (other than possibly re-scaling). This gives rise to a family of optimization problems:

   > Find a new distance metric d' which "best approximates" d under the condition that S and T have small earthmover distance under d',

   where we have the flexibility of choosing the measure of quality to how well d' approximates d.

Let M be the mapping of Equation 8.16. The following properties of M are easy to verify.

Proposition 8.4.1. *The mapping M defined in (8.16) satisfies*

1. *statistical parity between S and T up to bias ε,*
2. *the Lipschitz condition for every pair $(x, y) \in (S \times S) \cup (T \times T)$.*

Proof. The first property follows since

$$D_{\mathrm{tv}}(M(S), M(T)) = D_{\mathrm{tv}}\left(\mathop{\mathbb{E}}_{x \in S} \mathop{\mathbb{E}}_{y \sim \mu_x} \nu_y, \mathop{\mathbb{E}}_{x \in T} \nu_x\right) \leqslant D_{\mathrm{tv}}(\mu_S, U_T) \leqslant \varepsilon.$$

The second claim is trivial for $(x, y) \in T \times T$. So, let $(x, y) \in S \times S$. Then,

$$D(M(x), M(y)) \leqslant D(\mu_x, \mu_y) \leqslant d(x, y).$$

■

We have given up the Lipschitz condition between S and T, instead relying on the terms $d(x,y)$ in the objective function to discourage mapping x to distant y's. It turns out that the Lipschitz condition between elements $x \in S$ and $y \in T$ is still maintained on average and that the expected violation is given by $d_{\mathrm{EM+L}}(S,T)$ as shown next.

Proposition 8.4.2. *Suppose $D = D_{\mathrm{tv}}$ in (8.15). Then, the resulting mapping M satisfies*

$$\mathop{\mathbb{E}}_{x\in S} \max_{y\in T} \Big[D_{\mathrm{tv}}(M(x), M(y)) - d(x,y) \Big] \leqslant d_{\mathrm{EM+L}}(S,T).$$

Proof. For every $x \in S$ and $y \in T$ we have

$$\begin{aligned} D_{\mathrm{tv}}(M(x), M(y)) &= D_{\mathrm{tv}}\left(\mathop{\mathbb{E}}_{z\sim\mu_x} M(z), M(y) \right) \\ &\leqslant \mathop{\mathbb{E}}_{z\sim\mu_x} D_{\mathrm{tv}}\left(M(z), M(y)\right) && \text{(by Fact 8.2.6)} \\ &\leqslant \mathop{\mathbb{E}}_{z\sim\mu_x} d(z,y) && \text{(Proposition 8.4.1 since } z, y \in T) \\ &\leqslant d(x,y) + \mathop{\mathbb{E}}_{z\sim\mu_x} d(x,z) && \text{(by triangle inequalities)} \end{aligned}$$

The proof is completed by taking the expectation over $x \in S$. ∎

An interesting challenge for future work is handling preferential treatment of multiple protected subsets that are *not* mutually disjoint. The case of disjoint subsets seems easier and in particular amenable to our approach.

8.5 Small loss in bounded doubling dimension

The general LP shows that given an instance $\mathcal{I}$, it is possible to find an "optimally fair" mapping in polynomial time. The result however does not give a concrete quantitative bound on the resulting loss. Further, when the instance is very large, it is desirable to come up with more efficient methods to define the mapping.

We now give a fairness mechanism for which we can prove a bound on the loss that it achieves in a natural setting. Moreover, the mechanism is significantly more efficient than the general linear program. Our mechanism is based on the exponential mechanism [MT], first considered in the context of differential privacy.

We will describe the method in the natural setting where the mapping M maps elements of V to distributions over V itself. The method could be generalized to a different set A as long as we also have a distance function defined over A and some distance preserving embedding of V into A. A

natural loss function to minimize in the setting where V is mapped into distributions over V is given by the metric d itself. In this setting we will give an explicit Lipschitz mapping and show that under natural assumptions on the metric space (V, d) the mapping achieves small loss.

Definition 8.5.1. Given a metric $d\colon V \times V \to \mathbb{R}$ the exponential mechanism $\mathrm{E}\colon V \to \mu(V)$ is defined by putting

$$\mathrm{E}(x) \stackrel{\text{def}}{=} [Z_x^{-1} e^{-d(x,y)}]_{y \in V},$$

where $Z_x = \sum_{y \in V} e^{-d(x,y)}$.

Lemma 8.5.2 ([MT]). *The exponential mechanism is (D_∞, d)-Lipschitz.*

One cannot in general expect the exponential mechanism to achieve small loss. However, this turns out to be true in the case where (V, d) has small *doubling dimension*. It is important to note that in differential privacy, the space of databases does *not* have small doubling dimension. The situation in fairness is quite different. Many metric spaces arising in machine learning applications do have bounded doubling dimension. Hence the theorem that we are about to prove applies in many natural settings.

Definition 8.5.3. The *doubling dimension* of a metric space (V, d) is the smallest number k such that for every $x \in V$ and every $R \geqslant 0$ the ball of radius R around x, denoted $B(x, R) = \{y \in V : d(x, y) \leqslant R\}$ can be covered by 2^k balls of radius $R/2$.

We will also need that points in the metric space are not too close together.

Definition 8.5.4. We call a metric space (V, d) *well separated* if there is a positive constant $\varepsilon > 0$ such that $|B(x, \varepsilon)| = 1$ for all $x \in V$.

Theorem 8.5.5. *Let d be a well separated metric space of bounded doubling dimension. Then the exponential mechanism satisfies*

$$\mathop{\mathbb{E}}_{x \in V} \mathop{\mathbb{E}}_{y \sim \mathrm{E}(x)} d(x, y) = O(1).$$

Proof. Suppose d has doubling dimension k. It was shown in [CG] that doubling dimension k implies for every $R \geqslant 0$ that

$$\mathop{\mathbb{E}}_{x \in V} |B(x, 2R)| \leqslant 2^{k'} \mathop{\mathbb{E}}_{x \in V} |B(x, R)|, \tag{8.17}$$

where $k' = O(k)$. It follows from this condition and the assumption on (V, d) that for some positive $\varepsilon > 0$,

$$\mathop{\mathbb{E}}_{x \in V} |B(x, 1)| \leqslant \left(\frac{1}{\varepsilon}\right)^{k'} \mathop{\mathbb{E}}_{x \in V} |B(x, \varepsilon)| = 2^{O(k)}. \tag{8.18}$$

Then,

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\in V}\mathop{\mathbb{E}}_{y\sim \mathrm{E}(x)} d(x,y) &\leqslant 1 + \mathop{\mathbb{E}}_{x\in V}\int_1^\infty \frac{re^{-r}}{Z_x}|B(x,r)|\,\mathrm{d}r \\
&\leqslant 1 + \mathop{\mathbb{E}}_{x\in V}\int_1^\infty re^{-r}|B(x,r)|\,\mathrm{d}r \qquad (\text{since } Z_x \geqslant e^{-d(x,x)} = 1) \\
&= 1 + \int_1^\infty re^{-r}\mathop{\mathbb{E}}_{x\in V}|B(x,r)|\,\mathrm{d}r \\
&\leqslant 1 + \int_1^\infty re^{-r}r^{k'}\mathop{\mathbb{E}}_{x\in V}|B(x,1)|\,\mathrm{d}r \qquad (\text{using (8.18)}) \\
&\leqslant 1 + 2^{O(k)}\int_0^\infty r^{k'+1}e^{-r}\,\mathrm{d}r \\
&\leqslant 1 + 2^{O(k)}(k'+2)!.
\end{aligned}$$

As we assumed that $k = O(1)$, we conclude

$$\mathop{\mathbb{E}}_{x\in V}\mathop{\mathbb{E}}_{y\sim \mathrm{E}(x)} d(x,y) \leqslant 2^{O(k)}(k'+2)! \leqslant O(1).$$

■

Remark 8.5.6. If (V,d) is not well-separated, then for every constant $\varepsilon > 0$, it must contain a well-separated subset $V' \subseteq V$ such that every point $x \in V$ has a neighbor $x' \in V'$ such that $d(x,x') \leqslant \varepsilon$. A Lipschitz mapping M' defined on V' naturally extends to all of V by putting $M(x) = M'(x')$ where x' is the nearest neighbor of x in V'. It is easy to see that the expected loss of M is only an additive ε worse than that of M'. Similarly, the Lipschitz condition deteriorates by an additive 2ε, i.e., $D_\infty(M(x),M(y)) \leqslant d(x,y) + 2\varepsilon$. Indeed, denoting the nearest neighbors in V' of x,y by x',y' respectively, we have $D_\infty(M(x),M(y)) = D_\infty(M'(x'),M'(y')) \leqslant d(x',y') \leqslant d(x,y) + d(x,x') + d(y,y') \leqslant d(x,y) + 2\varepsilon$. Here, we used the triangle inequality.

The proof of Theorem 8.5.5 shows an exponential dependence on the doubling dimension k of the underlying space in the error of the exponential mechanism. The next theorem shows that the loss of any Lipschitz mapping has to scale at least linearly with k. The proof follows from a packing argument similar to that in [HT]. The argument is slightly complicated by the fact that we need to give a lower bound on the *average* error (over $x \in V$) of any mechanism.

Definition 8.5.7. A set $B \subseteq V$ is called an *R-packing* if $d(x,y) > R$ for all $x,y \in B$.

Here we give a lower bound using a metric space that may not be well-separated. However, following Remark 8.5.6, this also shows that any mapping defined on a well-separated subset of the metric space must have large error up to a small additive loss.

Theorem 8.5.8. *For every $k \geqslant 2$ and every large enough $n \geqslant n_0(k)$ there exists an n-point metric space of doubling dimension $O(k)$ such that any (D_∞, d)-Lipschitz mapping $M\colon V \to \mu(V)$ must satisfy*

$$\mathop{\mathbb{E}}_{x\in V} \mathop{\mathbb{E}}_{y\sim M(x)} d(x,y) \geqslant \Omega(k).$$

Proof. Construct V by randomly picking n points from a r-dimensional sphere of radius $100k$. We will choose n sufficiently large and $r = O(k)$. Endow V with the Euclidean distance d. Since $V \subseteq \mathbb{R}^r$ and $r = O(k)$ it follows from a well-known fact that the doubling dimension of (V,d) is bounded by $O(k)$.

Claim 8.5.9. *Let X be the distribution obtained by choosing a random $x \in V$ and outputting a random $y \in B(x,k)$. Then, for sufficiently large n, the distribution X has statistical distance at most $1/100$ from the uniform distribution over V.*

Proof. The claim follows from standard arguments showing that for large enough n every point $y \in V$ is contained in approximately equally many balls of radius k. ∎

Let M denote any (D_∞, d)-Lipschitz mapping and denote its error on a point $x \in V$ by

$$R(x) = \mathop{\mathbb{E}}_{y\sim M(x)} d(x,y).$$

and put $R = \mathbb{E}_{x\in V} R(x)$. Let $G = \{x \in V\colon R(x) \leqslant 2R\}$. By Markov's inequality $|G| \geqslant n/2$.

Now, pick $x \in V$ uniformly at random and choose a set P_x of 2^{2k} random points (with replacement) from $B(x,k)$. For sufficiently large dimension $r = O(k)$, it follows from concentration of measure on the sphere that P_x forms a $k/2$-packing with probability, say, $1/10$.

Moreover, by Claim 8.5.9, for random $x \in V$ and random $y \in B(x,k)$, the probability that $y \in G$ is at least $|G|/|V| - 1/100 \geqslant 1/3$. Hence, with high probability,

$$|P_x \cap G| \geqslant 2^{2k}/10. \tag{8.19}$$

Now, suppose M satisfies $R \leqslant k/100$. We will lead this to a contradiction thus showing that M has average error at least $k/100$. Indeed, under the assumption that $R \leqslant k/100$, we have that for every $y \in G$,

$$\mathbb{P}\{M(y) \in B(y, k/50)\} \geqslant \frac{1}{2}, \tag{8.20}$$

and therefore

$$\begin{aligned} 1 \geqslant \mathbb{P}\left\{M(x) \in \cup_{y \in P_x \cap G} B(y, k/2)\right\} &= \sum_{y \in P_x \cap G} \mathbb{P}\left\{M(x) \in B(y, k/2)\right\} \\ &\qquad \text{(since } P_x \text{ is a } k/2\text{-packing)} \\ &\geqslant \sum_{y \in P_x \cap G} \exp(-k)\, \mathbb{P}(M(y) \in B(y, k/2)) \\ &\qquad \text{(by the Lipschitz condition)} \\ &= \frac{2^{2k}}{10} \cdot \frac{\exp(-k)}{2} > 1\,. \end{aligned}$$

This is a contradiction which shows that $R > k/100$. ■

Open Question 8.5.1. Can we improve the exponential dependence on the doubling dimension in our upper bound?

8.6 Discussion and future Directions

In this chapter we introduced a framework for characterizing fairness in classification. The key element in this framework is a requirement that similar people be treated similarly in the classification. We developed an optimization approach which balanced these similarity constraints with a vendor's loss function. and analyzed when this local fairness condition implies statistical parity, a strong notion of equal treatment. We also presented an alternative formulation enforcing statistical parity, which is especially useful to allow preferential treatment of individuals from some group. Below we consider some open questions and directions for future work.

8.6.1 On the Similarity Metric

As noted above, one of the most challenging aspects of our work is justifying the availability of a distance metric. We argue here that the notion of a metric already exists in many classification problems, and we consider some approaches to building such a metric.

8.6.1.1 Defining a metric on individuals

The imposition of a metric already occurs in many classification processes. Examples include credit scores[1] for loan applications, and combinations of test

[1] We remark that the credit score is a one-dimensional metric that suggests an obvious interpretation as a measure of quality rather than a measure of similarity. When the metric is defined over multiple attributes such an interpretation is no longer clear.

scores and grades for some college admissions. In some cases, for reasons of social engineering, metrics may be adjusted based on membership in various groups, for example, to increase geographic and ethnic diversity.

The construction of a suitable metric can be partially automated using existing machine learning techniques. This is true in particular for distances $d(x, y)$ where x and y are both in the same protected set or both in the general population. When comparing individuals from different groups, we may need human insight and domain information. This is discussed further in Section 8.6.1.2.

Another direction, which intrigues us but which have not yet pursued, is particularly relevant to the context of on-line services (or advertising): allow users to specify attributes they do or do not want to have taken into account in classifying content of interest. The risk, as noted early on in this work, is that attributes may have redundant encodings in other attributes, including encodings of which the user, the ad network, and the advertisers may all be unaware. Our notion of fairness can potentially give a refinement of the "user empowerment" approach by allowing a user to participate in defining the metric that is used when providing services to this user (one can imagine for example a menu of metrics each one supposed to protect some subset of attributes). Further research into the feasibility of this approach is needed, in particular, our discussion throughout this chapter has assumed that a single metric is used across the board. Can we make sense out of the idea of applying different metrics to different users?

8.6.1.2 Building a metric via metric labeling

One approach to building the metric is to first build a metric on S^c, say, using techniques from machine learning, and then "inject" members of S into the metric by mapping them to members of S in a fashion consistent with observed information. In our case, this observed information would come from the human insight and domain information mentioned above. Formally, this can be captured by the problem of *metric labeling* [KT]: we have a collection of $|S^c|$ *labels* for which a metric is defined, together with $|S|$ objects, each of which is to be assigned a label.

It may be expensive to access this extra information needed for metric labeling. We may ask the question of how much information do we need in order to approximate the result we would get were we to have all this information. This is related to our next question.

8.6.1.3 How much information is needed?

Suppose there is an unknown metric d^* (the right metric) that we are trying to find. We can ask an expert panel to tell us $d^*(x,y)$ given $(x,y) \in V^2$. The experts are costly and we are trying to minimize the number of calls we need to make. The question is: How many queries q do we need to make to be able to compute a metric $d\colon V \times V \to \mathbb{R}$ such that the distortion between d and d^* is at most C, i.e.,

$$\sup_{x,y \in V} \max\left\{\frac{d(x,y)}{d^*(x,y)}, \frac{d^*(x,y)}{d(x,y)}\right\} \leqslant C\,. \tag{8.21}$$

The problem can be seen as a variant of the well-studied question of constructing *spanners*. A spanner is a small implicit representation of a metric d^*. While this is not exactly what we want, it seems that certain spanner constructions work in our setting as well, if we are willing to relax the embedding problem by permitting a certain fraction of the embedded edges to have arbitrary distortion, as any finite metric can be embedded, with constant slack and constant distortion, into constant-dimensional Euclidean space [ABC+].

8.6.2 Case study on applications in health care

An interesting direction for a case study is suggested by another Wall Street Journal article (11/19/2010) that describes the (currently experimental) practice of insurance risk assessment via online tracking. For example, food purchases and exercise habits correlate with certain diseases. This is a stimulating, albeit alarming, development. In the most individual-friendly interpretation described in the article, this provides a method for assessing risk that is faster and less expensive than the current practice of testing blood and urine samples. "Deloitte and the life insurers stress the databases wouldn't be used to make final decisions about applicants. Rather, the process would simply speed up applications from people who look like good risks. Other people would go through the traditional assessment process." [SM] Nonetheless, there are risks to the insurers, and preventing discrimination based on protected status should therefore be of interest:

> "The information sold by marketing-database firms is lightly regulated. But using it in the life-insurance application process would "raise questions" about whether the data would be subject to the federal Fair Credit Reporting Act, says Rebecca Kuehn of the Federal Trade Commission's division of privacy and identity protection. The law's provisions kick in when "adverse action" is taken against a person, such as a decision to deny insurance or increase rates."

We might also consider defining new protected sets, based on the presence of pre-existing conditions.

8.6.3 Does Fairness imply Privacy?

We have already discussed the need for privacy of (non-)membership in S in ensuring fairness. We now ask a converse question: Does fairness in the context of advertising ensure privacy?

Statistical parity has the interesting effect that it eliminates *redundant encodings* of S in terms of A, in the sense that after applying M, there is no $f : A \to \{0,1\}$ that can be biased against S in any way. This prevents certain attacks that aim to determine membership in S.

Unfortunately, this property is not hereditary. Indeed, suppose that the advertiser wishes to target HIV-positive people. If the set of HIV-positive people is protected, then the advertiser is stymied by the statistical parity constraint. However, suppose it so happens that the advertiser's utility function is extremely high on people who are not only HIV-positive but who also have AIDS. Consider a mapping that satisfies statistical parity for "HIV-positive," but also maximizes the advertiser's utility. We expect that the necessary error of such a mapping will be on members of "HIV\AIDS," that is, people who are HIV-positive but who do not have AIDS. In particular, we don't expect the mapping to satisfy statistical parity for "AIDS" – the fraction of people with AIDS seeing the advertisement may be much higher than the fraction of people with AIDS in the population as a whole. Hence, the advertiser can in fact target "AIDS".

Alternatively, suppose people with AIDS are mapped to a region $B \subset A$, as is a |AIDS|/|HIV positive| fraction of HIV-negative individuals. Thus, being mapped to B maintains statistical parity for the set of HIV-positive individuals, meaning that the probability that a random HIV-positive individual is mapped to B is the same as the probability that a random member of the whole population is mapped to B. Assume further that mappings to $A \setminus B$ also maintains parity. Now the advertiser can refuse to do business with all people with AIDS, sacrificing just a small amount of business in the HIV-negative community.

These examples show that statistical parity is not a good notion of privacy in targeted advertising. A natural question, not yet pursued, is whether we can get better privacy using the Lipschitz property with a suitable metric.

8.6.4 Related work

We briefly mention two related topics; to our knowledge the work in these fields does not solve the problems described here.

Learning. Fair classification bears some resemblance to learning. One difference, however, is that learning does not typically involve arbitrarily different costs for different classification errors, whereas our notion of loss does precisely this. Also, we tend not think in terms of a unique "correct" fair classification. Even if "correctness" makes sense when fairness is not taken into account, there may be many different ways to temper correctness in order to obtain fairness.

Fair Division (Cake-Cutting). As the name implies, fair classification also bears a resemblance to fair division, or "cake-cutting," in which the goal is to divide a resource in such a way that each recipient believes he or she has received a fair amount, as expressed in the individual's *utilities*. Certainly our goal of treating similar people similarly is grounded in the intuition that this is not only the right thing to do but also will help to reduce envy. However, traditional work in fair division has nothing corresponding to the vendor. Moreover, it is not at all clear that the vendor's notion of loss can be incorporated into the utilities of the individuals (the only degree of freedom available to us for trying to express additional goals), as our design goal requires fairness even in the presence of an arbitrary vendor. With an unreasonable vendor similar people could wind up with completely dissimilar utilities, and hence be treated differently.

8.6.5 Catalog of Evils

We briefly summarize here behaviors against which we wish to protect. We make no attempt to be formal. Let S be a protected set.

1. *Blatant explicit discrimination*. This is when membership in S is explicitly tested for and a "worse" outcome is given to members of S than to members of S^c.

2. *Discrimination Based on Redundant Encoding*. Here the explicit test for membership in S is replaced by a test that is, in practice, essentially equivalent.

3. *Redlining*. A well-known form of discrimination based on redundant encoding. The following definition appears in an article by [Hun1], which contains the history of the term, the practice, and its consequences: "Redlining is the practice of arbitrarily denying or limiting financial services to specific neighborhoods, generally because its residents are people of color or are poor."

4. *Cutting off business with a segment of the population in which membership in the protected set is disproportionately high.* A generalization of redlining, in which members of S need not be a majority of the redlined population; instead, the fraction of the redlined population belonging to S may simply exceed the fraction of S in the population as a whole.

5. *Self-fulfilling prophecy.* Here the vendor advertiser is willing to cut off its nose to spite its face, deliberately choosing the "wrong" members of S in order to build a bad "track record" for S. A less malicious vendor may simply select *random* members of S rather than qualified members, thus inadvertently building a bad track record for S.

6. *Reverse tokenism.* This concept arose in the context of imagining what might be a convincing refutation to the claim "The bank denied me a loan because I am a member of S." One possible refutation might be the exhibition of an "obviously more qualified" member of S^c who is also denied a loan. This might be compelling, but by sacrificing one really good candidate $c \in S^c$ the bank could refute all charges of discrimination against S. That is, c is a token rejectee; hence the term "reverse tokenism" ("tokenism" usually refers to accepting a token member of S). We remark that the general question of explaining decisions seems quite difficult, a situation only made worse by the existence of redundant encodings of attributes.

Bibliography

[ABC+] Ittai Abraham, Yair Bartal, Hubert T.-H. Chan, Kedar Dhamdhere, Anupam Gupta, Jon M. Kleinberg, Ofer Neiman, and Aleksandrs Slivkins. Metric embeddings with relaxed guarantees. In *FOCS*, pages 83–100. IEEE, 2005.

[AHK] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta algorithm and applications. Technical report, Princeton University, 2005.

[Ajt] Miklós Ajtai. $\Sigma^1[1]$-formulae on finite structures. *Ann. Pure Appl. Logic*, 24(1):1–43, 1983.

[BCD+] Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proc. 26th Symposium on Principles of Database Systems (PODS)*, pages 273–282. ACM, 2007.

[BCdWZ] Harry Buhrman, Richard Cleve, Ronald de Wolf, and Christof Zalka. Bounds for small-error and zero-error quantum algorithms. In *Proc. 40th Foundations of Computer Science (FOCS)*, pages 358–368. IEEE, 1999.

[BDMN] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the SuLQ framework. In *Proc. 24th Symposium on Principles of Database Systems (PODS)*, pages 128–138. ACM, 2005.

[BF] I. Barany and Z. Furedi. Approximation of the sphere by polytopes having few vertices. *Proceedings of the American Mathematical Society*, 102(3):651–659, 1988.

[BFJ+] Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael J. Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proc. 26th Symposium on Theory of Computing (STOC)*, page 262. ACM, 1994.

[BH] Maria-Florina Balcan and Nicholas J. A. Harvey. Learning submodular functions. In *Proc. 43nd Symposium on Theory of Computing (STOC)*, pages 793–802. ACM, 2011.

[BKN] Amos Beimel, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. In *Proc. 7th TCC*, pages 437–454. Springer, 2010.

[BLM1] S. Boucheron, G. Lugosi, and P. Massart. A sharp concentration inequality with applications. *Random Structures and Algorithms*, 16(3):277–292, 2000.

[BLM2] S. Boucheron, G. Lugosi, and P. Massart. On concentration of self-bounding functions. *Electronic Journal of Probability*, 14:1884–1899, 2009.

[BLR] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In *Proc. 40th Symposium on Theory of Computing (STOC)*, pages 609–618. ACM, 2008.

[BN] Hai Brenner and Kobbi Nissim. Impossibility of differentially private universally optimal mechanisms. In *FOCS*, pages 71–80. IEEE, 2010.

[Bou] J. Bourgain. On high dimensional maximal functions associated to convex bodies. *American Journal of Mathematics*, 108(6):1467–1476, 1986.

[BV] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[BZ] Michael Barbaro and Tom Zeller. A face is exposed for aol searcher no. 4417749. *New York Times*, 2006.

[CG] T.-H. Hubert Chan and Anupam Gupta. Approximating TSP on metrics with bounded global growth. In *Proc. 19th Symposium on Discrete Algorithms (SODA)*, pages 690–699. ACM-SIAM, 2008.

[CH] Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. In *Proc. 24th COLT*. Omnipress, 2011.

[Cha] Moses Charikar. Private communication. 2011.

[Che] Elliott W. Cheney. *Introduction to Approximation Theory*. McGraw-Hill, New York, New York, 1966.

[CKNZ] Chandra Chekuri, Sanjeev Khanna, Joseph Naor, and Leonid Zosin. A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM J. Discrete Math.*, 18(3):608–625, 2004.

[CM] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Proc. 22nd NIPS*, pages 289–296. MIT Press, 2008.

[Dal] Tore Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15:429–444, 1977.

[De] Anindya De. Lower bounds in differential privacy. *CoRR*, abs/1107.2183, 2011.

[DFK] Martin E. Dyer, Alan M. Frieze, and Ravi Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, 1991.

[DKM+] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Proc. 25th EUROCRYPT*, pages 486–503. Springer, 2006.

[DL] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proc. 41st Symposium on Theory of Computing (STOC)*, pages 371–380. ACM, 2009.

[DMNS] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. 3rd TCC*, pages 265–284. Springer, 2006.

[DMT] Cynthia Dwork, Frank McSherry, and Kunal Talwar. The price of privacy and the limits of LP decoding. In *Proc. 39th Symposium on Theory of Computing (STOC)*, pages 85–94. ACM, 2007.

[DN1] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proc. 22nd PODS*, pages 202–210. ACM, 2003.

[DN2] Cynthia Dwork and Kobbi Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Proc. 24th CRYPTO*, pages 528–544. Springer, 2004.

[DNPR] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proc. 42nd Symposium on Theory of Computing (STOC)*. ACM, 2010.

[DNR+] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil P. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proc*. 41*st Symposium on Theory of Computing (STOC)*, pages 381–390. ACM, 2009.

[DRV] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Proc*. 51*st Foundations of Computer Science (FOCS)*. IEEE, 2010.

[DWHL] Bolin Ding, Marianne Winslett, Jiawei Han, and Zhenhui Li. Differentially private data cubes: optimizing noise sources and consistency. In *SIGMOD Conference*, pages 217–228. ACM, 2011.

[Dwo1] Cynthia Dwork. Differential privacy. In *Proc*. 33*rd ICALP*, pages 1–12. Springer, 2006.

[Dwo2] Cynthia Dwork. The differential privacy frontier (extended abstract). In *TCC*, pages 496–502. Springer, 2009.

[Dwo3] Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, January 2011. Available from the author's web site.

[DY] Cynthia Dwork and Sergey Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In *Proc*. 28*th CRYPTO*, pages 469–480. Springer, 2008.

[Fel] Vitaly Feldman. A complete characterization of statistical query learning with applications to evolvability. In *Proc*. 50*th Foundations of Computer Science (FOCS)*, pages 375–384. IEEE, 2009.

[FFKN] Danny Feldman, Amos Fiat, Haim Kaplan, and Kobbi Nissim. Private coresets. In *Proc*. 41*st Symposium on Theory of Computing (STOC)*, pages 361–370. ACM, 2009.

[FRY] Stephen E. Fienberg, Alessandro Rinaldo, and Xiolin Yang. Differential privacy and the risk-utility tradeoff for multi-dimensional contingency tables. In *Privacy in Statistical Databases*, 2010.

[GCN+] George, Heeney Catherine, Hawkins Naomi, de Vries Jantina, Boddington Paula, Kaye Jane, Bobrow Martin, Weir Bruce, and P3G Consortium Church. Public access to genome-wide data: Five views on balancing research with privacy and protection. *PLoS Genet*, 5(10):e1000665, 10 2009.

[GH] Apostolos Giannopoulos and Marianna Hartzoulaki. Random spaces generated by vertices of the cube. *Discrete and Computational Geometry*, V28(2):255–273, 2002.

[GHIM] Michel X. Goemans, Nicholas J. A. Harvey, Satoru Iwata, and Vahab S. Mirrokni. Approximating submodular functions everywhere. In *Proc. 20th Symposium on Discrete Algorithms (SODA)*, pages 535–544. ACM-SIAM, 2009.

[GHRU] Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. Privately releasing conjunctions and the statistical query barrier. In *Proc. 43nd Symposium on Theory of Computing (STOC)*, pages 803–812. ACM, 2011.

[Gia] Apostolos Giannopoulos. Notes on isotropic convex bodies. Preprint, 2003.

[GLM+] Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private approximation algorithms. In *Proceedings of the Twenty First Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010. To appear.

[GLS] Martin Grötschel, Laszlo Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization (Algorithms and Combinatorics)*. Springer, December 1994.

[GMW+] Michaela Götz, Ashwin Machanavajjhala, Guozhang Wang, Xiaokui Xiao, and Johannes Gehrke. Publishing search logs - a comparative study of privacy guarantees. *TKDE*, 99(PrePrints), 2011.

[GRS] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In *Proc. 41st Symposium on Theory of Computing (STOC)*, pages 351–360. ACM, 2009.

[HLM] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially-private data release. *CoRR*, abs/1012.4763, 2010.

[HR] Moritz Hardt and Guy Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Proc. 51st Foundations of Computer Science (FOCS)*, pages 61–70. IEEE, 2010.

[HS] Lisa Hellerstein and Rocco A. Servedio. On PAC learning algorithms for rich boolean function classes. *Theoretical Computer Science*, 384(1):66–76, 2007.

[HSR+] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Stanley F. Nelson, and David W. Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS Genet*, 4(8):e1000167, 08 2008.

[HT] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proc. 42nd Symposium on Theory of Computing (STOC)*. ACM, 2010.

[Hun1] D. Bradford Hunt. Redlining. Encyclopedia of Chicago, 2005.

[Hun2] Neil Hunt. Netflix prize update. Netflix Blog (http://blog.netflix.com/2010/03/this-is-neil-hunt-chief-product-officer.html).

[Jac] Jeffrey C. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997.

[JKS] Jeffrey Jackson, Adam Klivans, and Rocco A. Servedio. Learnability beyond AC^0. In *Proc. 34th Symposium on Theory of Computing (STOC)*, pages 776–784. ACM, 2002.

[JM] Carter Jernigan and Behram F.T. Mistree. Gaydar: Facebook friendships expose sexual orientation. *First Monday*, 14(10), 2009.

[JPW] G. Jagannathan, K. Pillaipakkamnatt, and R.N. Wright. A Practical Differentially Private Random Decision Tree Classifier. In *2009 IEEE International Conference on Data Mining Workshops*, pages 114–121. IEEE, 2009.

[Kea] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

[KK] Bo'az Klartag and Gady Kozma. On the hyperplane conjecture for random convex sets. *Israel Journal of Mathematics*, 170(1):253–268, 2009.

[KKMN] Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. Releasing search queries and clicks privately. In *WWW*, pages 171–180. ACM, 2009.

[KKT] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM New York, NY, USA, 2003.

[Kla] Bo'az Klartag. On convex perturbations with a bounded isotropic constant. *Geometric and Functional Analysis (GAFA)*, 16(6):1274–1290, December 2006.

[KLN+] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? In *Proc*. 49*th Foundations of Computer Science (FOCS)*, pages 531–540. IEEE, 2008.

[KLS] Ravi Kannan, László Lovász, and Miklós Simonovits. Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. *Random Struct*. *Algorithms*, 11(1):1–50, 1997.

[KM1] David Kempe and Frank McSherry. A decentralized algorithm for spectral analysis. *Journal of Computer & System Sciences*, 74:70–83, 2008.

[KM2] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proc*. 30*th ACM SIGMOD*, pages 193–204. ACM, 2011.

[KM3] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *SIAM J*. *on Computing*, 22(6):1331–1348, 1993.

[KMR+] Michael J. Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. On the learnability of discrete distributions. In *Proc*. 26*th STOC*, pages 273–282. ACM, 1994.

[KOS] Adam Klivans, Ryan O'Donnell, and Rocco A. Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer & System Sciences*, 68(4):808–840, 2004.

[KRSU] Shiva Kasiviswanathan, Mark Rudelson, Adam Smith, and Jonathan Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *Proc*. 42*nd Symposium on Theory of Computing (STOC)*, pages 775–784. ACM, 2010.

[KS] Adam Klivans and Rocco A. Servedio. Learning DNF in time $2^{\widetilde{O}(n^{1/3})}$. *Journal of Computer & System Sciences*, 68(2):303–318, 2004.

[KT] Jon M. Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *Journal of the ACM (JACM)*, 49(5):616–639, 2002.

[LHR+] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. Optimizing linear counting queries under differential privacy. In *PODS*, pages 123–134. ACM, 2010.

[LPRTJ] Alexander E. Litvak, Alain Pajor, Mark Rudelson, and Nicole Tomczak-Jaegermann. Smallest singular value of random matrices and geometry of random polytopes. *Adv. Math.*, 195(2):491–523, 2005.

[LW] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.

[MM] Frank McSherry and Ilya Mironov. Differentially private recommender systems: building privacy into the net. In *Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 627–636. ACM, 2009.

[MP] Vitali D. Milman and Alain Pajor. Isotropic position and inertia ellipsoids and zonoids of the unit ball of a normed n-dimensional space. *Geometric Aspects of Functional Analysis*, 1376:64–104, 1989.

[MT] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proc. 48th Foundations of Computer Science (FOCS)*, pages 94–103. IEEE, 2007.

[Nao] Moni Naor. Evaluation may be easier than generation. In *Proc. 28th Symposium on Theory of Computing (STOC)*, pages 74–83. ACM, 1996.

[NRS] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proc. 39th Symposium on Theory of Computing (STOC)*, pages 75–84. ACM, 2007.

[NS] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy (S&P)*, pages 111–125. IEEE, 2008.

[RBHT] Benjamin I. P. Rubinstein, Peter L. Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for SVM learning. *CoRR*, abs/0911.5708, 2009.

[RR] Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *Proc*. 42*nd Symposium on Theory of Computing (STOC)*, pages 765–774. ACM, 2010.

[SA1] Emily Steel and Julia Angwin. On the web's cutting edge, anonymity in name only. *The Wall Street Journal*, 2010.

[SA2] Emily Steel and Julia Angwin. On the web's cutting edge, anonymity in name only. *The Wall Street Journal*, 2010.

[SF] Z. Svitkina and L. Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. In *Proc*. 49*th Foundations of Computer Science (FOCS)*, pages 697–706. IEEE, 2008.

[She] Alexander A. Sherstov. The intersection of two halfspaces has high threshold degree. In *Proc*. 50*th Foundations of Computer Science (FOCS)*. IEEE, 2009.

[SM] Leslie Scism and Mark Maremont. Insurers test data profiles to identify risky clients. *The Wall Street Journal*, 2010.

[Swe] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[UV] Jonathan Ullman and Salil P. Vadhan. PCPs and the hardness of generating private synthetic data. In *TCC*, pages 400–416. Springer, 2011.

[Val] Leslie Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Vem] Santosh Vempala. Geometric random walks: a survey. *MSRI Volume on Combinatorial and Computational Geometry*, 52:577–616, 2005.

[Von] J. Vondrak. A note on concentration of submodular functions. *Arxiv preprint arXiv:1005.2791*, 2010.

[WB] Samuel Warren and Louis Brandeis. The right to privacy. *Harvard Law Review*, 4(5), 1890.

[Wed] P.Å. Wedin. Perturbation bounds in connection with the singular value decomposition. *BIT*, 12:99–111, 1972.

[Zar] Tal Zarsky. Private communication. 2011.