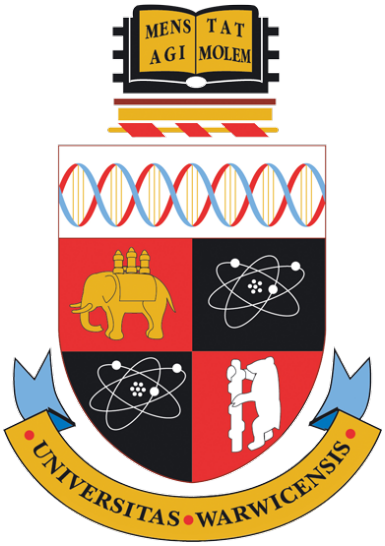
WARWICK
THE UNIVERSITY OF WARWICK



Efficient Estimation of Statistical Functions While Preserving Client-side Privacy

by

Tejas Kulkarni

Thesis

Submitted to the University of Warwick

in partial fulfilment of the requirements

for admission to the degree of

Doctor of Philosophy

Supervised by: Prof. Graham Cormode

Department of Computer Science

September 2019

Contents

Acknowledgments

I am grateful to **Prof. Graham Cormode** for providing me the opportunity to perform research under his supervision. I really appreciated how diligently he responded to even my silliest emails. I would like to express my gratitude to **Dr. Divesh Srivastava** for his time and valuable feedback to the drafts written by me and Graham. I will cherish my weekly Skype meetings with Graham and Divesh for a long time. Those discussions were a major source of learning for me at the University Of Warwick.

I would like to acknowledge the funding sources and facilities at the University of Warwick/Warwick Institute Of Science Of Cities (WISC) and the Alan Turing Institute in London that made my research work possible. I appreciate the caring support of the CDT administrator **Ms. Yvonne Colmer** for her prompt assistance in administrative matters. I am also thankful to **Neha Gupta**, a peer research student at WISC for her tremendous help from my early days in the UK.

I am beholden to my family – my parents (**Mr. Vijay Kulkarni** and **Mrs. Alka Kulkarni**), my younger brother (**Tanmay Kulkarni**) and my wife (**Amruta Kulkarni**). I could not have reached this far without their support.

Lastly, I feel indebted to the **faculties** of **Indian Institute Of Technology, Madras** where I previously studied for introducing me to academic research. It was the surreal environment there that inspired a mud clay like me to pursue researcher studies.

Declarations

I, Tejas Kulkarni, declare that the scholarly work presented in this thesis is led and performed by me during the doctoral studies at the Warwick Institute of Science Of Cities, Doctoral Training Centre in University Of Warwick. I have appropriately cited the sources where the information has been derived externally. This thesis has not been previously submitted in any form for a degree at Warwick or any other university.

I also declare that the thesis is less than 70,000 words in length, exclusive of tables and bibliographies.

Abstract

Aggregating service users' personal data for analytical purposes is a common practice in today's Internet economy. However, distrust in the data aggregator, data breaches and risks of subpoenas pose significant challenges in the availability of data. The framework of differential privacy is enjoying wide attention due to its scalability and rigour of privacy protection it provides, and has become a de facto standard for facilitating privacy preserving information extraction. In this dissertation, we design and implement resource efficient algorithms for three fundamental data analysis primitives, **marginal**, **range**, and **count** queries while providing strong differential privacy guarantees.

The first two queries are studied in the strict scenario of untrusted aggregation (aka local model) in which the data collector is allowed to only access the noisy/perturbed version of users' data but not their true data. To the best of our knowledge, marginal and range queries have not been studied in detail in the local setting before our works. We show that our simple data transfomation techniques help us achieve great accuracy in practice and can be used for performing more interesting analysis.

Finally, we revisit the problem of count queries under trusted aggregation. This setting can also be viewed as a relaxation of the local model called limited precision local differential privacy. We first discover certain weakness in a well-known optimization framework leading to solutions exhibiting pathological behaviours. We then propose more constraints in the framework to remove these weaknesses without compromising too much on utility.

Sponsorships and Grants

The work conducted in this Thesis has been sponsored by Marie Curie grant - 618202, Warwick Collaborative Postgraduate Research Scholarship, AT&T Labs, Warwick Institute for the Science of Cities, Department of Computer Science (The University Of Warwick), and Alan Turing Institute.

Acronyms

AM All Properies Mechanism for $\mathbb{L}_1, \mathbb{L}_2$.

AUC Area under the ROC Curve.

CH Column Honesty.

CI Constrained Inference.

CM Column Monotone.

DHT Discrete Haar Transform.

DP Differential Privacy.

EM Explicit Fair Mechanism/Expectation Maximization.

F Fair Mechanism.

FM Fair Mechanism for $\mathbb{L}_1, \mathbb{L}_2$.

FO Frequency Oracle.

GM Range Restricted Geometric Mechanism.

GRR Generalized Randomized Response.

HH$_B$ Hierarchical Histogram With Branching Factor B.

HRR Hadamard Randomized Response.

HT Hadamard Transform.

LDP Local Differential Privacy.

LLDP Limited Precision Local Differential Privacy.

LP Linear Programming.

MI Mutual Information.

MSE Mean Squared Error.

OLH Optimal Local Hashing.

OPT Optimal.

OUE Optimized Local Encoding.

PS Preferential Sampling.

RH Row Honesty.

RM Row Monotone.

RMSE Root Mean Squared Error.

ROC Receiver Operating Characteristic.

RR Randomized Response.

S Symmetric Mechanism.

UCM Unconstrained Mechanism for $\mathbb{L}_1, \mathbb{L}_2$.

UM Uniform Mechanism.

WH Weak Honesty.

WM Weak Honesty Mechanism.

List of Tables

List of Figures

> I first met Steve (Feinberg) during a talk I was giving at Carnegie Mellon in 2003 describing very early thoughts on a cryptography-flavored approach to privacy in public databases. Some of these ideas arose during Adam Smith's internship with me at Microsoft. Steve was critical ("Your utility is going to be in the toilet"), but I think he was intrigued by the cryptographic approach, since after the talk he proposed that we have a workshop ("Your bring your guys and I'll bring mine"). This occurred during the summer of 2005 in the hillside town of Bertinoro, Italy. The workshop almost broke down on the second day: the statisticians thought the cryptographers, with their talk of the adversary" and its arbitrary auxiliary information, were completely paranoid, while the cryptographers were frustrated by the absence of a formal notion of privacy and a measure of its loss in the statistical work. Fortunately, there is little to do in Bertinoro at night, other than to drink grappa in the piazza, and this eased the tension considerably. Later in the workshop Steve proposed to Alan Karr and me that we found a journal and, to paraphrase Gertrude Stein, we have and this is it.
>
> — Cynthia Dwork [1]

Chapter 1

Introduction

Large scale Internet based services have become ubiquitous and touch multiple facets of human life. Due to significant progress in the storage/processing technologies, machine learning science and exponential increase in the Internet penetration rate [2], these services are becoming increasingly capable of gathering and analyzing large *digital footprints* left intentionally/unintentionally by its users. This analysis is often used to distill valuable information about an individual user's service engagement bahaviors in order to improve the quality of her experience. Some examples include, streaming services like BBC iplayer [3], Youtube, Netflix, Spotify using user viewing/browsing history for promoting [4–6] new content; using location information reported by mobile devices to create interactive traffic maps and suggest less congested route [7]; digital telemetry systems [8–12] installed by various browser and app venders like Microsoft, Mozilla, Google, Safari and Snap to gather statistics.

While the utility of such large scale algorithmic data processing systems cannot be argued, the sheer rush to exploit on their potential have always presented huge challenges on the *data privacy* front that were largely ignored by the stakeholders for a long time. Time and again, careless data stewardship has resulted into high profile data breaches and unethical data usages including more recent Cambridge Analytica scandal [13] and Equifax data breach [14]. A holistic treatment of privacy risks maps across the disciplines. While some of the risks can be defined in a precise mathematical way, others require cross-disciplinary approaches.

1.1 Statistical Disclosure Techniques

To mitigate the privacy risks and increase availability of data, many *statistical disclosure limitation techniques* have been developed by the database/social science communities. These techniques were widely accepted by social scientists, statistical agencies for masking,

perturbing and generalizing the contributions of an individual in the dataset.

1.1.1 Data Anonymization

Anonymization removes personally identifiable information (e.g., social security number, name, address, and phone number) from the dataset to be released for research/analytics purposes with the intention of making it impossible for data consumers to identify the data participants whose privacy is under question. This simple approach was motivated from the following definition of privacy from Dalenius [15]. Dalenius in 1977 perceived privacy as a following goal of a database system: anything that can be learned from the database about a particular individual should be determined without the access to the database. The intuition behind this definition was to ensure that the change in the adversary's belief prior and posterior to a particular dataset's access remains small. However, satisfying this definition of privacy is not possible in the existence of unforeseeable background knowledge the adversary may possess. For example, in the late 1990s [16], Latanya Sweeney uniquely linked records in a de-identified dataset to identified records in publically available datasets and was able to disclose governor of Massachusetts's medical information.

1.1.2 K-Anonymity

Samarati and Sweeney [17] in the follow-up study proposed *k-anonymity* to address the drawbacks of simple anonymiziation scheme. A k-anonymized dataset satisfies that the property that every individual contained in the record is similar to at least another $k-1$ other records on the potentially identifying variables. $k-$anonymity is commonly achieved by two ways.

- **Generalization**: Replace the attribute values by a broader range in the category.
- **Suppression**: Anonymize centrain attributes by replacing those with a symbol e.g. $*$.

Machanavajjhala *et al.* [18] demonstrated that k-anonymity does not prevent adversary with arbitrary auxiliary information from improving his posterior knowledge significantly about an individual in case sensitive attributes are not diverse enough. Moreover, it doesn't offer privacy to a group. A k-anonymous database may reveal information about a group if that group is homogeneous with respect to some field.

1.1.3 l-Diversity

Since K-anonymity is vulnerable to inference attacks against sensitive attributes, Machanavajjhala [19] added additional layer of constraint on top of k-anonymity by proposing the notion of l-diversity which requires that each tuple that shares identical quasi-identifiers (a

set of attributes which could potentially identify an individual when used together) has at least l well-represented values for the sensitive attribute. Li *et al.* [18] showed that while l-diversity is robust against identity disclosure, it does not fully address the issue of attribute disclosure. In fact it is possible to link sensitive attributes to another. Moreover, when dataset has skewed distribution, perturbed and the original distributions could differ vastly.

1.1.4 t-closeness

Li *et al.* [18] refined the term l-diversity by adding another restriction. Li *et al.*proposed a threshold t to upper bound the statistical distance (often measured by the earth mover distance) between the distribution of the sensitive attribute values within an anonymized group as compared to the global distribution of values. While t-closeness provides privacy protection against attribute and identity disclosure attack, it may offer poor utility — meaning the statistical properties of the original dataset may be lost in the process of anonymization.

In summary, the defence mechanisms proposed in response to linking attacks including k-anonymity, l-diversity, and t-closeness suffer from one or more of the following shortcomings.

- Privacy protection offered was syntactic in nature i.e. property of anonymized dataset. Syntactic notions are typically protect against a particular inferencing strategy but adversary can consider other sources of information. Naturally, these notions were found to be vulnerable to more sophisticated linkage attacks in due course.
- Required identification of sensitive identifiers which is not always possible.
- Difficult to satisfy in case of multidimensional datasets.
- Provide unsatisfactory utility/accuracy.
- Privacy guarantees could only hold for datasets satisfying specific properties.
- Do not provide the worst case guarantees in the adversarial settings.

Understanding the limits of these techniques is still an area of active research.

Many de-anonymiziation attacks in wide range of domains: recommender systems [5, 20], social networks [21, 22], location data [23, 24], browsing history [25] rely on a simple principle — a small number of individually unidentifying data points about an individual can collectively identify that individual. In fact, there is a simple theoretical justification for this insight. The current world population of ~7.7 billion in 2019 can be represented by ~33 bits[1]. This means the worst case probability for adversary for identifying

[1]To the best of our knowledge, this interesting fact is known to be first mentioned by Arvind Narayanan on his blog `https://33bits.wordpress.com`

a person is $\frac{1}{2^{33}}$. Any additional information known about that person e.g. gender (represented by a bit) will reduce adversary's uncertainy by half. Knowing his/her city (with population 1M) will further reduce this probability to ~$\frac{1}{2^{12}}$.

1.2 Differential Privacy (DP)

The privacy guarantees of many privacy preserving disclosure technique were not comprehensive and rested on mere *absence* of known attacks. Repeated success of de-anonymization attacks made a strong case against these techniques and favoured frameworks providing formal mathematical guarantees. These attacks also showed that privacy techniques should also provide meaningful privacy protection in scenarios where arbitrary amounts of external information that may be available to adversary in the form of public datasets. This is because as long as an adversary has enough computing resources, the canonical approach of *hiding* an individual into a large enough crowd simply is insufficient as no dataset is large enough to limit the information disclosure when auxiliary information about an individual is available.

The paradigm of DP [26] developed concurrently along with some of the previous methods stood-out compared to its predecessors/contemporaries due to its robustness to a wide range of attacks (including those that are not even known at the time of deployment) and most importantly, it provides guarantees regardless of the adversary's background knowledge by introducing a parameter ϵ (a.k.a. privacy budget) that captures the tension between privacy and utility/accuracy. Informally, a randomized algorithm (a.k.a mechanism) satisfying DP perturbs a dataset by adding controlled noise in such a way that the algorithm produces similar outcomes irrespective of a specific individual's participation in that dataset. In other words, the worst case change in a DP compliant algorithm's outcome after adding/removing an individual's record remains bounded. An immediate consequence of this property is that when an adversary is presented with an outcome of a differentially private algorithm, his degree of uncertainty on whether a specific individual's record was included in the dataset remains bounded irrespective of his background knowledge about that specific individual. This means that individual is protected almost as if his/her information is excluded from the analysis. A similar rationale holds when considering privacy of a *group*, however, at a degraded privacy guarantee.

After more than a decade long rigourous development, DP is being accepted as a gold standard for privacy in academic communities. Several organizations e.g. US Census Bureau [27], Apple [28], Microsoft [9], Google [10, 29], Uber [30] have already deployed products that use DP to privately aggregate telemetry data or generate synthetic data from their private databases. In fact, DP is attracting interest in legal communities also and many recent research/position articles including [31–33] recommend DP as a method of choice for

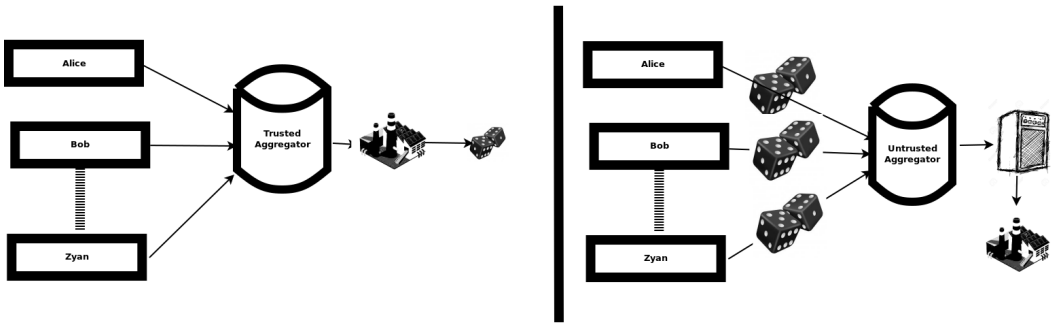


Figure 1.1: Difference between trusted (centralized) and untrusted (local) aggregation in case of DP.

designing privacy regulations (e.g. EU GDPR[2], HIPAA[3]) compliant data analysis systems.

Untrusted Aggregation. While the original model of DP assumes that the data curator (a.k.a. aggregator) is trusted and has access to the entire dataset, in many realistic settings the data participants do not trust the aggregator and may not be comfortable in sharing their sensitive data with the aggregator. Besides, collecting and storing large amounts of true user data imposes undue risks at aggregator's end which s/he may not be willing to take. This is why a variant of DP known as *local* DP [34, 35] (LDP) has risen to prominence in recent years. Under LDP, individuals retain control of their own private data, by revealing only (noisy) randomized transformations of their input. Aggregating the reports of sufficiently many users gives accurate answers, while preserving each individual's privacy. This creates a win-win situation for all parties involved. In fact, DP's widespread industrial adoption can be largely attributed to the recent development in the LDP techniques. Figure 1.1 shows the difference between trusted and untrusted aggregation under DP. Note that in contrast to LDP, the data perturbation step under traditional/centralized DP model happens at the aggregator's end. We focus most of this dissertation on facilitating untrusted aggregation of basic statistics under local differential privacy guarantees.

1.3 Research Questions Of Interest

Motivated by the scenarios of *untrusted aggregation*, that are becoming increasingly commonplace (also due the increased scrutiny by regulatory bodies/legislation's), this dissertation mainly focuses on *developing novel algorithms for various statistical problems while providing client side privacy*. We address following research problems in this dissertation.

Marginal Queries. Many analysis and machine learning tasks require the availability of joint distribution/marginal statistics on multidimensional datasets while providing strong

[2]General Data Protection Regulation
[3]Health Insurance Portability and Accountability

privacy guarantees for the data subjects. Applications for these statistics range from finding correlations in the data to learning sophisticated prediction models. We provide a set of algorithms for materializing marginal statistics under LDP. We prove the first tight theoretical bounds on the accuracy of marginals compiled under each approach, perform empirical evaluation to confirm these bounds, and evaluate them for tasks such as modeling and correlation testing. Our results show that releasing information based on (local) Fourier transformations of the input is preferable to alternatives based directly on (local) marginals.

Range Queries. Counting the fraction of a population having an input within a specified interval i.e. a range query, is a fundamental data analysis primitive. A simple baseline mechanism is to aggregate a histogram privately using existing primitives and simply add the counts in the relevant cells. This baseline provides highly unsatisfactory accuracy for mid/longer sized range queries since the error in aggregation grows linearly with the interval size. We describe and analyze two classes of approaches for range queries, based on the hierarchical histograms and the Haar wavelet transform. We show that both have strong theoretical accuracy guarantees on error. In practice, both methods are fast and optimal in communication. Our experiments show that the wavelet approach is most accurate in high privacy settings, while the hierarchical approach dominates for weaker privacy requirements. To the best of our knowledge, ours is among the first non-industrial work to provide simulations with domain sizes as large as 2^{22}.

Count Queries. The local model requires the randomized transformation of each item to be indistinguishable from every other item's transformation in the output space of the algorithm. This requirement could be too strong or even unnecessary in some cases. It may be possible to design more accurate algorithms for such settings if we relax the original definition of LDP. Towards this goal, we focus on the core problem of count queries and design mechanisms to release data associated with a group of n individuals. Prior works [36, 37] consider formulating this problem as an optimization problem and use linear programs to obtain a mechanism for a target objective function (e.g. avg./worst case error). However, solving such optimization may have undesired consequences, leading to yielding mechanisms producing pathological behaviours in practice. We eliminate these behaviours by suggesting additional constraints in the linear programs. We demonstrate in a set of experiments on real and synthetic data which is preferable in practice, for different combinations of data distributions, constraints, and privacy parameters. Though we consider the centralized DP model in this work for simplicity, the propositions of this contributions have found more applications in the recently proposed relaxation of LDP [38].

Challenges. Considering these problems entails tackling several challenges, including:

1. Dealing with a highly restricted algorithmic design space. Most carefully thought algorithms for data collection in the trusted aggregation settings exploit the fact the

curator has access to the entire dataset. This is not the case under untrusted aggregation environments. Each user can see only his/her data and aggregator can access only the perturbed dataset.

2. Making these algorithms as resource efficient as we can since the devices (e.g. smartphones) on which the client DP mechanisms are deployed are often resource constrainted i.e. have limited computational power, storage and bandwidth.

3. Providing formal mathematical privacy and utility guarantees for the developed algorithms and confirming these with rigourous experimental evaluation on real/synthetic datasets.

Potential Use cases. The main aim of this dissertation is to facilitate privacy preserving aggregation of basic statistics regarding discrete valued datasets. The first two of our solutions can easily find applications in the scenarios in which cloud based service providers with potentially millions of subscribers intending to collect statistics about the activity of their users and their client-side software. For example, a smartphone app based food delivery service may be interested in understanding the most frequently ordered food combinations from its users. Similarly, an online video content provider may want to log the genre type of content users consume over a time period. In both the cases, user inputs can be encoded as sparse vectors. These providers could promote new content/menu items by offering simple "people who watched/ordered this also watched/ordered that" type suggestions. Towards this goal, estimation of joint/marginal distributions of subsets of the features (chapter 4) is an important prerequisite. The same video content service is capable of logging online traffic related data (e.g number of users served/connections open every second) at high precisions and intends to privately compute the fraction of load during a festive week handled by their servers to optimize on resource allocation. A solution to this problem boils down to answering range queries (chapter 5) over large time domains.

In another scenario, the food delivery service also computed the number of female customers and would like to share a differentially private version of number with a third party analytics company. Employing mechanisms presented in chapter 6 for releasing count data would help achieve more utility for the same level of privacy under certain conditions over baseline methods.

1.4 Thesis Contributions

The high level goal of this dissertation is to design and evaluate differentially private mechanisms for answering various statistical queries while providing client side privacy in case of untrusted aggregation. The first two algorithmic contributions satisfy local differential privacy.

1.5 Publications

Most work in this thesis has been led by me and performed in collaboration with Graham Cormode and Divesh Srivastava. The three main chapters are based on the following research articles. All authors are arranged in alphabetical order in these articles.

1.5.1 Main Publications

1. Answering Range Queries Under Local Differential Privacy [39], ACM Very Large Databases (ACM VLDB), 2019
2. Marginal Release Under Local Differential Privacy Model [40], In Proceedings of the International Conference on Management of Data (ACM SIGMOD), 2018
3. Constrained Private Mechanisms for Count Data [41, 42], In Proceedings Of IEEE International Conference on Data Engineering (IEEE ICDE), 2018, IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), 2019

1.5.2 Additional Contributions

1. Differentially Private Distributed Computation of U-Statistics [43], To appear in International Conference on Artificial Intelligence and Statistics (AISTATS), 2020
2. Privacy at Scale: Local Differential Privacy in Practice [44], Co-presented a tutorial at ACM SIG Knowledge Discovery And Data Mining (KDD), 2018. The slides can be found at `https://sites.google.com/view/kdd2018-tutorial/home`

1.6 Thesis Structure

This thesis is organized as follows.

- Chapter 2 provides some background about various database queries, differential privacy, relevant privacy preserving mechanisms and finally some machine learning/statistics concepts.
- Chapter 3 then surveys general prior work in local differential privacy and then specific to the context of database queries of interest.
- Next, chapters 4, 5, and 6 based on publications [39–42] respectively contain the technical contributions related to marginal, range, and count queries.
- Finally, chapter 7 concludes this dissertation by summarizing our findings and proposing few potential future directions and limitations of our study.

Chapter 2

Technical Preliminaries

This chapter summarizes various notions and technical tools used throughout this dissertation.

2.1 Terminologies

2.1.1 Client/User

The main goal of this dissertation is to develop privacy-preserving mechanisms for facilitating data collection and crowdsourcing of various basic statistics.

Definition 1. *(Client/User). A client/user is an electronic device participating in the data aggregation process that holds an individual's data.*

We assume that the clients have sufficient computing, networking, and storage capabilities to run mechanisms proposed in the dissertation. We also assume that the clients are non-colluding and may not be aware of each other's presence in the network. Since we intend to cover a broad range of contexts, a client could be any device from a smartphone to an orbiting satellite.

2.1.2 Databases

Definition 2. *(Database). A database $\mathcal{D}$ is a multiset of N structured records with each record belonging to an individual drawn from a fixed domain.*

Definition 3. *(Database Query). A query is a computational function applied to a database.*

Often database records are viewed to be logically/physically stored and maintained in a single location by a trusted curator. However, in the upcoming chapters, we also consider a decentralized setting in which the database is distributed across individuals and they can only access their own records. The curator in this case is not trustworthy and cannot access or query the database records. We use the term database interchangeably with dataset throughout this dissertation.

2.1.3 Aggregator/Server

Definition 4. *(Aggregator). An aggregator/server is an entity (a person or an organization) that collects data from the clients for analysis and further consumption purpose.*

In our case, aggregator is not trustworthy and/or obliged on legal/ethical grounds to collect client data only in the perturbed format. We assume that privacy mechanisms running at client/aggregator sides along with the parameters used is common knowledge. However, aggregator cannot see client's private coin tosses.

2.2 Database Queries Of Interest

2.2.1 Range Query

A range query $R_{[a,b]}$ counts the fraction of population N with their inputs within the range $[a, b]$. More formally, for N individuals each with an item $\{z_i \in [D]\}_{i=1}^{N}$ and $a < b, a \in [D], b \in [D]$, a range query $R_{[a,b]} \geq 0$ is to compute

$$R_{[a,b]} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_{a \leq z_i \leq b}$$

where $\mathbb{I}_p$ is a binary variable that takes the value 1 if the predicate p is true and 0 otherwise. Often the problem of computing/approximating range queries entails finding an efficient datastructure that allows faster/succinct answers to queried intervals. Multidimensional range queries are addressed by geometric data structures such as k-d trees or quadtrees [45]. As the dimension increases, these methods suffer from the *curse of dimensionality*, and it is usually faster to simply scan the data. This dissertation focuses on answering one dimensional queries which itself is a challenging problem to begin with under LDP.

Prefix Query. Prefix queries form an important class of range queries, where the left end of the interval is fixed. We only consider the prefix queries with the left end fixed to the first item in the domain i.e. 0.

Quantile Queries. Prefix queries are sufficient to answer quantile queries. The ϕ-quantile for $\phi \in [0, 1]$ is the index j in the domain such that at most a ϕ-fraction of the input data lies below j, and at most a $(1 - \phi)$ fraction lies above it. If we can pose arbitrary prefix queries, then we can binary search for a prefix j such that the prefix query on j meets the ϕ-quantile condition.

2.2.2 Marginal Query

Put simply, a marginal query involving k attributes returns the table with the joint (empirical) probability distribution for all combinations of those k-attributes. Thus, the contingency,

Attribute	Explanation
CC	Has customer paid using credit card?
Toll	Has customer paid toll?
Far	Is journey distance $\geq$ 10 miles?
Night_pick	Is pickup time $\geq$ 8 PM?
Night_drop	Is drop off time $\leq$3 AM?
M_pick	Is trip origin within Manhattan?
M_drop	Is trip destination within Manhattan?
Tip	Is tip paid $\geq$ 25% of the total fare?

Table 2.1: Attributes of NYC taxi dataset.

Trip/Attributes	M_pick	M_drop	CC	Tip	...
1	Y	N	N	Y	...
2	N	Y	Y	N	...
⋮	⋮	⋮	⋮	⋮	⋮
⋮	Y	N	Y	N	...

Table 2.2: A sample trip data.

or marginal, query is the workhorse of data analysis. These statistics are important in and of themselves for understanding the data distribution, and identifying which attributes are correlated. They are also used for query planning and approximate query answering within database systems. A variety of fundamental inference and machine learning tasks also rely on accurate marginals capturing the correlations. E.g. many algorithms in statistical language modeling/predictive text [46] and association rule mining (market basket analysis) compute low order marginals as a preprocessing step. Furthermore, for multivariate distributions where direct sampling is in-feasible or too costly, low dimensional marginals serve as building blocks [47, 48] to compute accurate approximations.

M_pick/M_drop	Y	N
Y	0.55	0.15
N	0.10	0.20

Table 2.3: An example of a 2-way marginal.

Motivating example: movement patterns. Consider the collection and release of statistics on movement patterns of individual's. Table 2.2 shows an example taxi trip dataset, where each journey is described in terms of a number of (binary) attributes including origin and destination, timings, tip, and mode of payment. Table 2.3 shows a sample marginal table consisting of two attributes which confirms that most trips are short and originated and

terminated within the Manhattan region and shows strong degree of correlation between pickup and drop off locations.

Notations and preliminaries. We restrict our attention to marginals involving binary attributes. We assume that each user's record comprises of d binary attributes. It is often more convenient to view the user's data instead as an indicator vector z_i of length $D = 2^d$ with 1 at exactly one place j_i and 0's at remaining positions. We model each user i's bit vector $z_i \in \mathcal{I}_{2^d \times 2^d}$ as a vertex in a d-dimensional Hamming cube. This representation is also called as the *one hot encoding* and captures *correlation* between multiple attributes of z_i. Then we can restrict our attention only on a subset of k dimensions of interest by summing (marginalizing) out cells of non-essential dimensions. This is formally captured by the following definition.

Definition 5 (Marginal operator)**.** *Given a vector $z \in \mathbb{R}^{2^d}$, the* marginal operator $\mathcal{C}^\beta : \mathbb{R}^{2^d} \Rightarrow \mathbb{R}^{2^k}$ *computes the summed frequencies for all combinations of values of attributes encoded by $\beta \in \{0,1\}^d$, where $|\beta|$, the number of* 1*s in β, is $k \leq d$.*

For example, for $d = 4$ and $\beta = 0101$ (which encodes our interest in the second and the fourth attribute), the result of $\mathcal{C}^{0101}(z)$ is the projection of t on all possible combinations of the second and fourth attributes with remaining attributes marginalized out. Each of the 2^k entries in the vector $\mathcal{C}^{0101}(z)$ stores the total frequency of combinations of the k attributes identified by β. We make use of the $\preceq$ relation, defined as $\alpha \preceq \beta$ iff $\alpha \wedge \beta = \beta$. For convenience of expression, we abuse notation and allow $C^\beta(z)$ to be indexed by $\{0,1\}^d$ rather than $\{0,1\}^k$, with the convention that entries α such that $\alpha \not\preceq \beta$ are 0. Under this indexing, the entries in a marginal can be written in the following way:

$$\forall \gamma \preceq \beta \quad C^\beta(z)[\gamma] = \textstyle\sum_{\eta : \eta \wedge \beta = \gamma} z[\eta] \tag{2.1}$$

The condition $\eta \wedge \beta = \gamma$ selects all indices $\eta \in \{0,1\}^d$ whose value on attributes encoded by β are γ.

Example 2.2.1. *Let $d = 4$ and $\beta = 0101$. Then, applying* (2.1)*:*

$$C^{0101}(z)[0000] = z[0000] + z[0010] + z[1000] + z[1010]$$
$$C^{0101}(z)[0001] = z[0001] + z[0011] + z[1001] + z[1011]$$
$$C^{0101}(z)[0100] = z[0100] + z[0110] + z[1100] + z[1110]$$
$$C^{0101}(z)[0101] = z[0101] + z[0111] + z[1101] + z[1111]$$

All indices in $\{0,1\}^d$ contribute exactly once to one entry in C^{0101}.

Definition 6 (k-way marginals)**.** *We say that β identifies a k-way marginal when $|\beta| = k$. For a fixed k, the set of all k-way marginals correspond to all $\binom{d}{k}$ distinct ways of picking k attributes from d. We refer to the set of* full *k-way marginals as encompassing all j-way marginals sets, $\forall j \leq k$.*

2.3 Differential Privacy (DP)

As mentioned briefly in Section 1.2, the framework of DP provides a more comprehensive way of expressing and quantifying the notion of privacy than most of its predecessors/contemporaries.

2.3.1 Definitions

Consider a database $\mathcal{D}$ in which each record $z_i \in \mathcal{D}$ is contributed by an individual $i \in [N]$. Each $z_i \in Domain(D)$ may consists of a fixed set of attributes.

Definition 7. *(Neighborhood.) Two datasets $\mathcal{D}, \mathcal{D}'$ of size N are in the neighborhood/adjacent to each other if the following holds.*

$$\exists!(z_i \neq z_i', z_i \in \mathcal{D}, z_i' \in \mathcal{D}'), i \in [N]$$

i.e. $\mathcal{D}, \mathcal{D}'$ are neighbors if they differ by exactly one record.

Definition 8. *(Pure Differential Privacy [49].) A randomized mechanism $\mathcal{M}$ is differentially private if the following holds for every pair of neighboring datasets $\mathcal{D}, \mathcal{D}'$.*

$$\exp(-\epsilon) \leq \frac{\Pr[O = \mathcal{M}(D)]}{\Pr[O = \mathcal{M}(D')]} \leq \exp(\epsilon), O \in Range(\mathcal{M})$$

The definition of DP states that the ratio of probabilities outputting the response for two neighboring datasets are bounded. This means the risk of information leakage born by an individual is *bounded* whether or not s/he participates in the dataset. Therefore, that individual can be assured that his/her participation would not drastically change the outcome of the analysis. The notion of a participant's inclusion and exclusion is encoded by the definition of neighborhood and the bound is measured by the parameter ϵ. Larger the ϵ, more the influence of an individual's record on the analysis resulting into a weaker guarantee. Similarly, smaller ϵ alludes to a smaller degree of sensitivity caused by any changes in an individual's record to $\mathcal{M}$'s outputs. Since $\mathcal{D}, \mathcal{D}'$ can be switched interchangeably, the ratio can be bound from the left side also.

A variant called approximate DP also allows a mechanism $\mathcal{M}$ to fail with a small probability $\delta > 0$.

Definition 9. *(Approximate Differential Privacy [49].) A randomized mechanism $\mathcal{M}$ is (ϵ, δ)-differentially private if the following holds for every pair of neighboring datasets $\mathcal{D}, \mathcal{D}'$.*

$$\Pr[O = \mathcal{M}(D)] \leq \exp(\epsilon) \times \Pr[O = \mathcal{M}(D')] + \delta, O \in Range(\mathcal{M})$$

δ is set to a cryptographically small constant $\delta \leq \frac{1}{N^{\omega(1)}}$ in a dataset of size N. In this thesis, we only design mechanisms satisfying pure DP.

A more nuanced understanding the definition can be gained by understanding the capabilities of the adversary DP promises to defend against.

Threat Model. Assume a two party game between a trusted curator and an omniscient adversary (with unbounded computing power). The curator draws $\mathcal{D}$ arbitrarily. The adversary has the following knowledge.

- $\mathcal{M}, \epsilon$ and the coins tossed by $\mathcal{M}$.
- $\mathcal{D}_{\neg i}$, i.e. $\mathcal{D}$ *excluding a record* z_i *sampled randomly*.
- $O \in Range(\mathcal{M})$, for $O(domain(\mathcal{D}))$ queries of $\mathcal{M}(D)$.

With these resources at disposal, adversary's goal is to predict z_i. The adversary's best guess for z_i record is

$$\underset{z_i \in Domain(\mathcal{D})}{\arg\max} \Pr[\mathcal{M}(\mathcal{D}_{\neg i} \cup z_i) = O]$$

In other words, the adversary should pick z_i that maximizes the probability of producing O which he can do by executing $\Pr[\mathcal{M}(\mathcal{D}_{\neg i} \cup z_i)]$ for $O(domain(\mathcal{D}))$ number of times. However, since the ratio between any probabilities for any pair neighboring datasets is bounded, adversary's probability of correctly guessing z_i is also limited. This means, for a small ϵ, it is nearly impossible for adversary to figure out changes made in a single record.

Next we study some important properties of DP.

Definition 10. *(Composition Property [50].) Consider a set* $\{\mathcal{M}_j\}_{j=1}^{K}$ *of DP compliant mechanisms each providing* ϵ_j*-DP.*

1. *Any cascading/composition of* $\mathcal{M}_j$*'s on a dataset* $\mathcal{D}$ *satisfies* $\sum_{j=1}^{K} \epsilon$*-DP.*
2. *Any cascading/composition of* $\mathcal{M}_j$*'s over K* partitions *of* $\mathcal{D}$*'s satisfies* $max_{j \in [K]} \epsilon_j$*-DP.*

Reasoning about the privacy guarantee of an intricate DP algorithm can be challenging. The composition property enables us to design DP mechanisms in a modular fashion.

Definition 11. *(Post-processing [50].) Given a* ϵ*-DP compliant mechanism* $\mathcal{M} : X \implies Y$ *and any function* $F : Y \implies Z$*, the composition* $F(\mathcal{M}(.))$ *is also* ϵ*-DP compliant.*

This property suggests that the transformation made by a DP algorithm are not lost even after further processing and the guarantee remains intact.

Definition 12. *(Global Sensitivity [50].) For a function F, pth norm $\mathbb{L}_p$ and all neighboring pairs $\mathcal{D}, \mathcal{D}'$, the global sensitivity is defined as follows.*

$$\Delta_p F = max_{\mathcal{D},\mathcal{D}'} ||F(\mathcal{D}) - F(\mathcal{D}')||_p$$

The global sensitivity of a query function F measures the maximum change in the outcome of F, a pair of neighboring datasets can cause. Differentially private mechanisms work by perturbing the dataset with statistical noise. To mask identity of each participant, it is often necessary to add noise calibrated by the sensitivity of the intended query function. Therefore, a lot of work is usually dedicated in discovering clever ways of bounding the sensitivity of the query function at hand.

2.3.2 Perturbation primitives satisfying ϵ-DP

Laplace Mechanism. One of the simplest mechanisms to perturb numeric attributes with additive noise is the zero mean Laplace mechanism. The zero mean Laplace distribution with scale b has the density function $\Pr[z; b] = \frac{1}{2b} \exp(\frac{-|z|}{b})$.

Theorem 1. *For any query function F with output $O \in \mathbb{R}^k, k \geq 1$, the Laplace mechanism $\mathcal{M}(\mathcal{D}) = F(\mathcal{D}) + \langle L_1, L_2, .., L_k \rangle$ satisfies ϵ-DP, where $L_1, .., L_k$ are i.i.d Laplace random variables with density $Lap(\frac{\Delta_1 F}{\epsilon})$.*

Exponential Mechanism [51]. McSherry and Talwar in [51] proposed the Exponential Mechanism as a generic approach to design mechanisms. Let $\mathbb{D}$ be the domain of input dataset and $\mathcal{R}$ the range of perturbed responses. The crux of the exponential mechanism is in designing a *quality function* $Q : \mathbb{D} \times \mathcal{R} \Longrightarrow \mathbb{R}$ so that $Q(d, r)$ measures the desirability of providing output r for input d. This mechanism is particularly suitable when queries are non numeric and we are required to encode our preference for output r when input is d. The mechanism is then defined by setting

$$\Pr[r \in \mathcal{R}|d] = \exp\left(\frac{\epsilon Q(d,r)}{2\Delta_p Q}\right) \Big/ \sum_{r' \in \mathcal{R}} \exp\left(\frac{\epsilon Q(d,r')}{2\Delta_p Q}\right) \tag{2.2}$$

where $\Delta_p Q$ is the global sensitivity of function Q i.e. the amount by which changing an individual's input can alter the output of Q in the worst case. It is proved that this mechanism obtains at least $\exp(\frac{-\epsilon}{2})$-differential privacy.

2.4 Local Differential Privacy

Initial work on differential privacy assumed the participation of a trusted aggregator, who curates the private information of individuals, and releases information through a DP algorithm. In practice, individuals may be reluctant to share private information with the

central data curator. Local differential privacy instead captures the case when each user independently (but collaboratively) releases information on their input through an instance of a DP algorithm. The original input never leaves user's end.

Now we introduce this model more formally. In the simplest setting, we have N non-colluding data-owners and each participant $i \in [N]$ has a private input z_i drawn from some global discrete or continuous distribution θ over a domain $\mathcal{Z}$. Any two tuples z_i and z_i' are considered adjacent, with $||z_i - z_{i'}||_1 \leq 2$. Implicitly, there is also an (untrusted) aggregator interested in estimating some statistics over the private dataset $\{z_i\}_{i=1}^N$.

Local Differential Privacy (LDP) [34, 35]. A randomized function $\mathcal{M}$ is ϵ-locally differentially private if for all possible pairs of $z_i, z_i' \sim \mathcal{Z}$ and for every possible output tuple O in the range of $\mathcal{M}$:

$$\exp(-\epsilon) \leq \frac{\Pr[O = \mathcal{M}(z_i)]}{\Pr[O = \mathcal{M}(z_i')]} \leq \exp(\epsilon), O \in Range(\mathcal{M})$$

In this local instantiation of DP, $\mathcal{M}$ is applied to each input independently. In contrast to the centralized model, perturbation under LDP happens at the user's end. We now compare the centralized and the local model in more detail.

- **Neighborhood Definition.** In LDP, all items in the input domain are neighbors of each other whereas in the centralized setting, all datasets differing by a single record are neighbors of each other.

- **Sensitivity Calculation.** It is not required in the local case due to modified neighborhood definition.

- **Perturbation Method.** In contrast to the centralized case, only input perturbation is allowed in LDP since user's data are not allowed to leave its end.

- **Error In Estimation.** The main consequence of input perturbation in LDP is significant increase in noise level. Specifically, works including [52, 53] proved that for any LDP compliant mean estimation mechanism $\mathcal{M}$ along with a post-processing function f estimates the true mean μ with at least following error.

$$\left| \frac{\sum_{i=1}^N f(\mathcal{M}(z_i))}{N} - \mu \right|_\infty = \Omega(\frac{1}{\epsilon\sqrt{N}})$$

On the other hand, this lower bound is $\frac{1}{\epsilon N}$ in the centralized case [54]. A similar bound exists for histogram aggregation. This means quadratically more number of users are required in the local case to match the accuracy level of the centralized model.

- **Algorithm Design.** Compared to the centralized case, the algorithm design space is limited in the local setting since the original dataset is not available. In fact, many tasks that can be performed in the centralized setting are impossible to perform in the local model with acceptable utility [34].

- **Interactive Algorithms.** In the local model, it is possible to design algorithms involving multiple rounds of data collection. Each round however consumes some amount of privacy budget since intermediate results are also required to satisfy LDP. In the centralized case, accessing the dataset multiple times may not cost privacy budget since only the final result is required to satisfy the DP guarantee. However, careful accounting of privacy budget consumed is still needed in the both the cases.

- **Resource Trade-off.** As a consequence of previous factors, any LDP algorithm designed should take into account, the communication/storage/computation trade-off since these algorithms are often deployed on resource constrainted devices. Communication overhead may not even exist in the centralized model. Managing storage/processing trade-off is a relatively less important challenge compared to the local model since perturbation and processing happens at aggregator's side.

Standard perturbation primitives satisfying LDP. Research questions in LDP have mostly focused around developing primitives for answering simple statistical queries such as mean and histogram. The primitives proposed are used as building blocks in designing solutions to more complex problems. In what follows, we will discuss some perturbation mechanisms developed for mean and histogram. For mean estimation, we restrict our attention to only the binary inputs case.

2.4.1 Mean Estimation

The simplest mechanism satisfying LDP, 1 bit randomized response [55] (RR) was proposed much before the theory of DP was even built. Here, we consider a setting with N participants each having a single bit $z_i \in \{1, 0\}$ of private information (an answer to a sensitive question). The aggregator's goal is estimate the approximation $\widehat{f}$ of the true fraction f of people satisfying with input as 1.

Pertubation. Each user reports his true answer z_i with probability p. Otherwise, s/he returns a bit drawn uniformly at random. In order for this mechanism to satisfy LDP, we want that

$$\frac{\Pr[1|1]}{\Pr[1|0]} = \frac{\Pr[0|0]}{\Pr[0|1]} = \frac{p}{1-p} \leq \exp(\epsilon)$$

Since the worst case ratio is $\exp(\epsilon)$, we replace the inequality by equality. Thus, by setting $p = \frac{\exp(\epsilon)}{\exp(\epsilon)+1}$, this simple mechanism satisfies ϵ-LDP.

Aggregation. Let O be the fraction of perturbed bits $\{z_i^*\}_{i=1}^N$ received by aggregator. We have

$$O = \frac{\sum_{i=1}^N z_i^*}{N} = pf + (1-p)(1-f)$$

Rearranging the expression yields us a frequency estimator $\widehat{f} = \frac{O-p+1}{2p-1}$.

Lemma 1. *$\widehat{f}$ is an unbiased estimator of f i.e. $\mathbb{E}[\widehat{f}] = f$.*

Proof.

$$\mathbb{E}[\widehat{f}] = \mathbb{E}[\frac{O-p+1}{2p-1}] = \frac{pf + (1-p)(1-f) - p + 1}{2p-1} = \frac{-p(1-f) - (1-p)(1-f) + 1}{2p-1}$$
$$= \frac{(1-f)(1-2p)+1}{2p-1} = f - 1 + 1 = f$$

□

Often the accuracy in estimation is measured via the mean squared error $\mathbb{E}[(\widehat{f} - f)^2]$. We know that the mean squared error can be expressed as sum of variance and bias. Since estimation is unbiased, mean squared error is determined by variance. Now we measure the variance in estimation.

Lemma 2. *The scaled variance* $\mathrm{Var}[\widehat{f} - f] = \frac{\exp(\epsilon)}{(\exp(\epsilon)-1)^2}$.

Proof.

$$\mathrm{Var}[\widehat{f} - f] = \mathrm{Var}[\widehat{f}] = \mathrm{Var}[\frac{O-p+1}{2p-1}] = \frac{\mathrm{Var}[O]}{(2p-1)^2}$$

We know that the observed fraction O consists of two Bernoulli distributions with parameters p and $1-p$ for fractions f and $1-f$. Therefore,

$$\frac{\mathrm{Var}[O]}{(2p-1)^2} = \frac{fp(1-p) + (1-f)(1-p)p}{(2p-1)^2} = \frac{p(1-p)}{(2p-1)^2} = \frac{\frac{\exp(\epsilon)}{1+\exp(\epsilon)} \cdot \frac{1}{1+\exp(\epsilon)}}{(\frac{2\exp(\epsilon)}{1+\exp(\epsilon)} - 1)^2} \tag{2.3}$$

$$= \frac{\exp(\epsilon)}{(\exp(\epsilon)-1)^2} \tag{2.4}$$

□

An asympotically similar expression can be derived for the case when user i holds $z_i \in \{-1, 1\}$ instead of $\{1, 0\}$.

As we will see in the next chapters, this classical idea is at the heart of many industrial deployments. In fact, there are multiple variants of this general principle. For

more involved mechanisms and usecases, the survey book [56] compiled by Chaudhari and Mukerjee can be referred.

2.4.2 Point Queries and Frequency Oracles (FO)

A basic question in the LDP model is to answer *point queries* on the distribution: to estimate the frequency of any given element z from the domain $\mathcal{Z}$. Answering such queries form the underpinning for a variety of applications such as population surveys, machine learning, spatial analysis and, as we shall see, our objective of quantiles and range queries.

In the point query problem, each i holds a private item z_i drawn from a public set $\mathcal{Z} = \{0, .., D-1\} = [D]$ using an unknown common discrete distribution θ. That is, θ_z is the probability that a randomly sampled input element is equal to $z \in \mathcal{Z}$. The goal is to provide a protocol in the LDP model (i.e. steps that each user and the aggregator should follow) so the aggregator can estimate θ as $\widehat{\theta}$ as accurately as possible. Solutions for this problem are referred to as providing a *frequency oracle*.

Several variant constructions of frequency oracles have been described in recent years. In each case, the users perturb their input locally and send the result to the aggregator. These noisy reports are aggregated and an appropriate bias correction is applied to them to reconstruct the frequency for each item in $\mathcal{Z}$. The error in estimation is once again quantified by the *variance* since often estimators for these mechanisms are *unbiased* and have the same variance for all items in the input domain. The mechanisms vary based on their computation and communication costs, and the accuracy (variance) obtained. The most practical mechanisms have the *unscaled* variance of $\mathcal{O}\left(\frac{\exp(\epsilon)}{N(\exp(\epsilon)-1)^2}\right)$. Some of these mechanisms were proposed nearly concurrently in multiple communities such as data management, privacy/security and information theory with different names.

Randomized Response On Input/Optimal Unary Encoding (INPRR/OUE) [57, 58]. It is also possible to use a single bit randomized response to aggregate histograms. Once again we assume that each user i's input $z_i \in \mathcal{I}_{D\times D}$, where $\mathcal{I}_{D\times D}$ is the set of all one hot encoded/identity basis vectors. The mechanism involves following transformation.

Perturbation. The bit at each $j \in [D]$ is perturbed as follows.

$$\Pr[z_i^*[j] = 1] = \begin{cases} p, \text{ if } z_i[j] = 1 \\ q, \text{ if } z_i[j] = 0 \end{cases}$$

Each perturbed vector z_i^* is sent to the aggregator. Note that in contrast with the version discussed in Section 2.4.1, two separate probabilities are used in this perturbation. Intuitively, since the index where the bit is 1 stores the original input, we expect p to as high as possible. On the other hand, we want fewer 0's to be flipped to 1's. Therefore, q should

be as low as possible. The LDP ratio is maximized when the numerator/denominator is maximized/minimized i.e.

$$\exp(\epsilon) = \frac{p(1-q)}{q(1-p)}$$

Rearranging gives $p = \frac{q\exp(\epsilon)}{\exp(\epsilon)q-q+1}$.

Aggregation. The aggregator collects the noisy bit vectors z_i^* for all users and adds them to the noisy histogram $T^* \in [D]$. Similar to Section 2.4.1, for each item $j \in [D]$ be the observed fraction of 1's is

$$T^*[j] = pf_j + (1-f_j)q$$

The estimated fraction f_j of item j is $\widehat{f_j} = \frac{T^*[j]-q}{p-q}$.

Lemma 3. *The scaled variance* $\mathrm{Var}[\widehat{f_j}] = \mathcal{O}\Big(\frac{\exp(\epsilon/2)}{(1+\exp(\epsilon/2))^2}\Big), \forall j \in [D]$

Proof.

$$\mathrm{Var}[\widehat{f_j} - f_j] = \mathrm{Var}[\widehat{f_j}] = \mathrm{Var}\Big[\frac{T^*[j]-q}{p-q}\Big] = \frac{\mathrm{Var}[T^*[j]]}{(p-q)^2} = \frac{p(1-p)f_j + (1-f_j)q(1-q)}{(p-q)^2}$$
$$= \frac{f_j(p-q)[1-p+q] + q(1-q)}{(p-q)^2} = \frac{f_j[1-p+q]}{p-q} + \frac{q(1-q)}{(p-q)^2}$$

For small f_j, which is often the case when D gets large, the variance is dominated by the second term. Therefore, $\mathrm{Var}[\widehat{f_j}] \approx \frac{q(1-q)}{(p-q)^2}$. Plugging p gives,

$$\mathrm{Var}[\widehat{f_j}] \approx \frac{q(1-q)}{(\frac{q\exp(\epsilon)}{\exp(\epsilon)q-q+1} - q)^2} = \frac{((\exp(\epsilon)-1)q+1)^2}{(\exp(\epsilon)-1)^2 q(1-q)} \tag{2.5}$$

□

Next we have two choices.

1. p and q are symmetric i.e. $p+q=1$. This simply leads to $p = \frac{\exp(\epsilon/2)}{1+\exp(\epsilon/2)}$ and $q = \frac{1}{1+\exp(\epsilon/2)}$. The variance becomes

$$\mathrm{Var}[\widehat{f_j}] \approx \frac{\exp(\epsilon/2)}{(\exp(\epsilon/2)-1)^2} \tag{2.6}$$

2. p and q are not symmetric. In this case, we can attempt to obtain the values of p and q

that minimize 2.5. By differentiating the variance with q, we get

$$\frac{\partial \mathrm{Var}[\widehat{f_j}]}{\partial q} = \frac{\partial\Big(\frac{((\exp(\epsilon)-1)q+1)^2}{(\exp(\epsilon)-1)^2 q(1-q)}\Big)}{\partial q} = \frac{1}{(\exp(\epsilon)-1)^2}\Big[\frac{\exp(\epsilon^2)}{(1-q)^2} - \frac{1}{q^2}\Big]$$

Setting the gradient above to 0 and solving for q produces $p = \frac{1}{2}$ and $q = \frac{1}{\exp(\epsilon)+1}$ And the minimum variance is

$$\mathrm{Var}[\widehat{f_j}] \approx \frac{4\exp(\epsilon)}{(\exp(\epsilon)-1)^2} \tag{2.7}$$

Though the case of asymmetric p and q appears to be more genetic and provides an expression for the minimum variance, the difference between 2.6 and 2.7 is not significant for the most widely used values of ϵ. While OUE/INPRR is simple to implement and provides best possible variance, it does not scale well to very large D due to large communication complexity (i.e., D bits from each user), and the consequent computation cost for the user ($\mathcal{O}(D)$ time to flip the bits).

Generalized Randomized Response(GRR)/Preferential Sampling (INPPS) [57–59]. It turns out that randomized response can be extended easily to categorical inputs.

Perturbation. Each input z_i reports the true value with probability p and with probability $1-p$, s/he reports $z_i' \neq z_i$ sampled uniformly at random from $[D]$. The LDP ratio is maximized when

$$\exp(\epsilon) = \frac{\Pr[j|j]}{\Pr[j|l]} = \frac{p}{\frac{1-p}{D-1}}, \forall j, l \in [D], j \neq l$$

Solving for p gives $p = \frac{\exp(\epsilon)}{\exp(\epsilon)+D-1}$.

Aggregation. The aggregator populates a noisy histogram $T^* \in \mathbb{R}^D$ upon collecting all noisy responses.

$$T^*[j] = pf_j + \sum_{k\in\{[D]\setminus j\}} \frac{f_k(1-p)}{D-1} = pf_j + \frac{(1-p)(1-f_j)}{D-1}, \forall j \in [D]$$

The true fraction f_j for each item $j \in [D]$ is estimated as

$$\widehat{f_j} = \frac{(D-1)T^*[j] + p - 1}{(Dp-1)}, \forall j \in [D]$$

The main shortcoming of this method is that the probability of outputting the truth decreases rapidly as D increases. For example, for $\epsilon = 1.1$ and $D = 100$ the truth probability p is only 0.02. This means, most of the times the mechanism reports a false random value. Let's

verify this observation by computing the variance.

Lemma 4. *The scaled variance* $\mathrm{Var}[\widehat{f}_j] = \mathcal{O}\Big(\frac{D\exp(\epsilon)}{(\exp(\epsilon)-1)^2}\Big), \forall j \in [D]$

Proof.

$$\begin{aligned}
\mathrm{Var}[\widehat{f}_j - f_j] = \mathrm{Var}[\widehat{f}_j] &= \mathrm{Var}\Big[\frac{(D-1)T^*[j]+p-1}{(Dp-1)}\Big] = \Big(\frac{D-1}{Dp-1}\Big)^2 \mathrm{Var}\Big[T^*[j]\Big] \\
&= \Big(\frac{D-1}{Dp-1}\Big)^2 \Big[p(1-p)f_j + \frac{p(1-p)(1-f_j)}{D-1}\Big] \\
&= \Big(\frac{D-1}{Dp-1}\Big)^2 p(1-p)\Big[f_j + \frac{(1-f_j)}{D-1}\Big] \\
&= \Big(\frac{D-1}{\frac{D\exp(\epsilon)}{\exp(\epsilon)+D-1}-1}\Big)^2 \Big(\frac{\exp(\epsilon)}{\exp(\epsilon)+D-1}\Big)\Big(\frac{D-1}{\exp(\epsilon)+D-1}\Big)\Big(\frac{f_j(D-2)+1}{D-1}\Big) \\
&= \Big(\frac{D-1}{\frac{(D-1)(\exp(\epsilon)-1)}{\exp(\epsilon)+D-1}}\Big)^2 \Big(\frac{\exp(\epsilon)[f_j(D-2)+1]}{(\exp(\epsilon)+D-1)^2}\Big) = \frac{\exp(\epsilon)[f_j(D-2)+1]}{(\exp(\epsilon)-1)^2} \\
&\leq \frac{\exp(\epsilon)(D-1)}{(\exp(\epsilon)-1)^2}
\end{aligned}$$

Where the last inequality comes by upper bounding $f_j \leq 1$. □

As we can see, the variance grows linearly with D, which is an undesirable feature for any mechanism. Nevertheless, Wang *et al.* in [57] showed that GRR/INPPS is among the best approaches when $D < 3\exp(\epsilon) + 2$. In chapter 4, we provide alternate proofs to OUE/ INPRR and INPPS/GRR.

Optimal Local Hashing (OLH) [57]. Wang *et al.* [57] proposed the OLH mechanism to deal with prohibitive communication cost. OLH aims to reduce the impact of dimensionality on accuracy by employing *universal hash functions*[1]. More specifically, each user samples a hash function $H : [D] \to [g]$ ($g \ll D$) u.a.r from a universal family $\mathbb{H}$ and perturbs the hashed input.

Perturbation. User i samples a $H_i \in \mathbb{H}$ u.a.r (principle D) and computes $h_i = H_i(v_i)$. User i then perturbs $h_i \in [g]$ using GRR. Specifically, each user reports H_i and, with probability $p = \frac{e^\epsilon}{e^\epsilon+g-1}$ gives the true h_i, else she reports a value sampled u.a.r from $[g]$.

Aggregation. The aggregator collects the perturbed hash values from all users. For each hash value h_i, the aggregator computes a frequency vector for all items in the original domain, based on which items would produce the hash value h_i under H_i. All N such histograms are added together to give $T^* \in \mathbb{R}^D$ and an unbiased estimator for each frequency for all

[1]A family of hash functions $\mathbb{H} = \{H : [D] \to [g]\}$ is said to be universal if $\forall z_i, z_j \in [D], z_i \neq z_j$: $\Pr_{H\in\mathbb{H}}[H(z_i) = H(z_j)] \leq \frac{1}{g}$ i.e. collision probability behaves uniformly.

elements in the original domain is given by the correction

$$\widehat{f_j} = (T^*[j] - \frac{1}{g})/(p - \frac{1}{g})$$

For $g = \exp(\epsilon) + 1$, OLH achieves the same variance as OUE/INPRR (i.e. $\frac{4\exp(\epsilon)}{(\exp(\epsilon)-1)^2}$) and yet at more economical communication of $\mathcal{O}(\log(D))$ bits per user compared to D bits for OUE/INPRR. However, a major downside is that it is compute intensive in terms of the decoding time at the aggregator's side, which is prohibitive for very large dimensions (say, for D above tens of thousands), since the time cost is proportional to $\mathcal{O}(ND)$.

2.4.3 Limited Precision LDP

The definition of LDP requires a randomized mechanism to satisfy the same limited information disclosure guarantee on *every pair of inputs* in the domain of the mechanism. This constraint can be too strong or even unnecessary or even insufficient in some contexts. For example, consider users with sensitive documents and the aggregator wants to collect histogram of word frequencies. Due to the large domain size of all possible words in a language and heavy tailed distribution for word counts, the amount of noise added can overwhelm the signal in the data.

While one line of research focuses on designing mechanisms tailored for massive domain sizes, another proposes alterations in the original definition of LDP. Schein *et al.* [38] proposed one such generalization of LDP for private Bayesian inference.

Definition 13. ***Limited Precision LDP (LLDP) [38].*** *A randomized function $\mathcal{M}$ is (k, ϵ)-limited precision locally differentially private if for pairs of $z_i, z_i' \sim \mathcal{Z}$ such that $||z_i - z_i'||_1 \leq k$ and for every possible output tuple O in the range of $\mathcal{M}$:*

$$\exp(-\epsilon) \leq \frac{\Pr[O = \mathcal{M}(z_i)]}{\Pr[O = \mathcal{M}(z_i')]} \leq \exp(\epsilon), O \in Range(\mathcal{M})$$

When $||z_i||_1 \leq k, \forall z_i \in \mathcal{Z}$, $(k, \epsilon)-$LLDP implies ϵ-LDP. This version was originally introduced in the context of centralized DP by Flood *et al.* [60] to protect portfolios of large firms. Andrés *et al.* [61] later extended it to protect location information.

Note that a (k, ϵ)-LLDP mechanism satisfies centralized ϵ-DP with global sensitivity k. Therefore, at times it is convenient to consider the centralized setting while designing LLDP algorithms.

Geometric Mechanism (GM) [36]. Ghosh *et al.* in their seminal work [36] proposed a discrete equivalent of Laplace mechanism for count queries. They consider a simple *trusted aggregation* with n participants each having a single bit of private information. A trusted aggregator intends to develop a randomized mechanism $\mathcal{M}$ for releasing the sum of n bits

privately.

Definition 14. ***Range Restricted Geometric Mechanism (GM) [36].*** *Let q be the true (unperturbed) result of a count query. The GM responds with $\min(\max(0, q+\delta), n)$, where δ is a noise drawn from a random variable X with a double sided geometric distribution, $\Pr[X=\delta] = \frac{(1-\alpha)\alpha^{|\delta|}}{1+\alpha}$ for $\delta \in \mathbb{Z}$.*

where $\alpha = \exp(\frac{-\epsilon}{\Delta_1})$, and Δ_1 is the global sensitivity of the query function under $\mathbb{L}_1$ norm. The global sensitivity Δ_1 for count queries is 1 since flipping an individual's bit can only change the sum by 1. The GM adds noise from two sided geometric distribution to the query result and remaps all outputs less than 0 onto 0 and greater than n to n. In fact it is possible to find the closed form solution for GM.

Linear Programming Framework [36]). Ghosh *et al.* developed a linear programming framework for designing mechanisms optimizing on a loss function for count queries. These mechanisms can be viewed as a column/row stochastic matrix with every i, jth entry being the probability $\Pr[i|j]$ of outputting i for input j. We describe their framework below.

Ghosh *et al.* 's key observation was that the DP requirements can be written as linear constraints over variables which represent the entries of the mechanism. The objective function is also a linear function of these variables. Formally, we define variables $\rho_{i,j}$ for $\Pr[i|j]$, and write:

$$\text{minimize:} \quad \sum_{j=0}^{n} w_j \sum_{i=0}^{n} |i-j|^p \rho_{i,j} \tag{2.8}$$

$$\text{subject to:} \quad 0 \leq \rho_{i,j} \leq 1 \quad \forall i, j \in [n] \tag{2.9}$$

$$\sum_{i=0}^{n} \rho_{i,j} = 1 \quad \forall j \in [n] \tag{2.10}$$

$$\rho_{i,j} \geq \alpha \rho_{i,j+1}, \text{ and } \rho_{i,j+1} \geq \alpha \rho_{i,j} \quad \forall i \in [n], j \in [n-1] \tag{2.11}$$

The constraints can be understood as follows: (2.9), (2.10) ensure that the entries of the matrix are probabilities and each column encodes a probability distribution, i.e. sums to 1. Constraint (2.11) encodes the differential privacy constraints. Finally, (2.8) encodes a loss function (cf. Definition 21) for the notion of utility we aim for. Common choices for p are 0, 1 and 2 corresponding to $\mathbb{L}_0, \mathbb{L}_1$ and $\mathbb{L}_2$ norms.

2.5 Statistics

In this section, we review some concepts that will be used in the later sections.

2.5.1 Chow-Liu Trees

Approximating a high dimensional joint distribution with d discrete variables is a classic problem in statistics. The main motivation for approximation comes from difficulty (or at times in-feasibility) in computing a large number of conditional and marginal probabilities involved in the joint distribution.

Bayesian networks are often used to represent conditional dependencies in a high dimensional distribution using a directed acyclic graph. It is possible to approximate a joint distribution as a product of multiple second order marginal and conditional distributions. For example, the joint distribution illustrated by the variable dependency network in figure 2.1(a) can be approximated in 2.1(b) as follows.

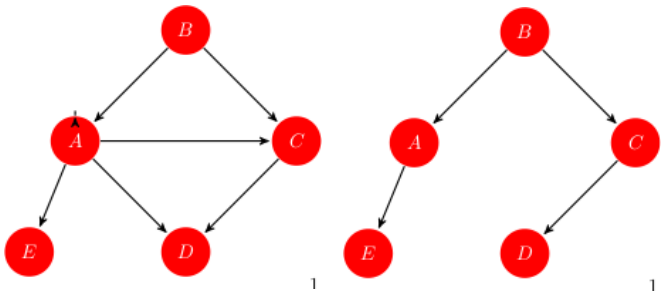
$$\Pr[B, A, C, D, E] \approx \Pr[B] \Pr[A|B] \Pr[C|B] \Pr[D|C] \Pr[E|A]$$

In the equation above, each probability is conditioned on at most one variable. Therefore, the problem of finding a second order approximation to a high dimensional distribution reduces to approximating a directed acyclic graph with a directed tree that optimizes a particular distance metric. Chow and Liu in [47] proved that a tree configuration that maximizes the *total mutual information* among edges is an optimal approximation of the joint distribution in question. This insight converts the intractable optimization problem of finding such tree to an easy problem of finding a maximum weight spanning tree. Concretely, all we have to do is treat all random variables as nodes in an empty graph and find a tree that maximizes the total edge weight. Once a tree is learnt, any high dimensional joint distribution of interest can be learnt by multiplying conditional probabilities that can found using marginals. The centre piece of this algorithm is computation of mutual information between $\binom{d}{2}$ pairs of variables. Mutual information between two discrete variables $A, B \in \{0, 1\}$ is given as

$$MI(A, B) = \sum_{i,j \in \{0,1\}^2} \Pr[A = i, B = j] \log \frac{\Pr[A = i, B = j]}{\Pr[A = i] \Pr[B = j]}$$

2.5.2 Association Testing

We often want to check if two variables A, B are independent or not i.e. we want to know if $\Pr[A, B] \approx \Pr[A] \Pr[B]$. The χ^2 test of independence compares the observed cell counts to expected counts assuming the independence (null hypothesis) and compute the χ^2 value (see e.g. [62] then compares this value to the critical value p for a given confidence



(a) Dependency Graph (b) Approximated Tree

Figure 2.1: Approximation of a variable dependency network using trees.

interval (usually 0.95). If $\chi^2 > p$, we conclude that A, B are dependent (rejecting the null hypothesis). For a 2-way marginal m involving binary random variables, the χ^2 statistic is

$$\sum_{j \in \{0,1\}^2} \frac{(m[j] - \mathbb{E}[m[j]])^2}{\mathbb{E}[m[j]]}$$

where $\mathbb{E}[m[j]]$ is the expected value at $m[j]$.

2.5.3 Area under the ROC Curve

In binary classification with class imbalance, the *Receiver Operating Characteristic (ROC)* gives the true positive rate with respect to the false positive rates of a predictor at each possible decision threshold. The Area under the ROC Curve (AUC) [63] is a popular summary of the ROC curve which gives a single, threshold-independent measure of the classifier quality corresponding to the probability that the predictor assigns a higher score to a randomly chosen positive point than to a randomly chosen negative one. Formally, let $\mathcal{X} \subset \mathbb{R} \times \{-1, 1\}$ and $\mathcal{S} = \{(z_i, y_i)\}_{i=1}^N$ where for each data point i, $z_i \in \mathbb{R}$ is the score assigned to point i and $y_i \in \{-1, 1\}$ is its label. For convenience, let $\mathcal{S}^+ = \{z_i : y_i = 1\}$ and $\mathcal{S}^- = \{z_i : y_i = -1\}$ and let $N^+ = |\mathcal{S}^+|$ and $N^- = |\mathcal{S}^-|$. The AUC is given by

$$AUC = \frac{1}{N^+ N^-} \sum_{z_i \in \mathcal{S}^+} \sum_{z_j \in \mathcal{S}^-} \mathbb{I}_{z_i > z_j}. \tag{2.12}$$

where $\mathbb{I}_\sigma$ is an indicator variable with value 1 if the predicate σ is true and 0 otherwise.

Chapter 3

Related Work

In this chapter, we initially discuss prior works on the local differential privacy. Then we restrict our attention to literature on the marginal, range and count queries. To the best of our knowledge, since the problems of marginal and range queries have not been studied before our attempts in the local DP settings, we only discuss prior treatment in the context of the centralized model for those problems. We are aware that differentially private untrusted aggregation can also be facilitated by combining DP with other contemporary technologies including secure multiparty aggregation [64] and trusted execution environment [65], we restrict our scope to only technologies involving LDP mechanisms.

3.1 Local Differential Privacy

The model of local differential privacy has risen in popularity in recent years in theory and in practice as a special case of differential privacy. This model was first suggested by Evfimievski *et al.* [66] under the name of γ-amplification, with an application to mining association rules. Duchi *et al.* [35] studied a generalization of that model as a local version of DP, and proposed a minimax framework with information theoretic bounds on utility. Recently, a substantial amount of effort has been put into the question of collecting simple popularity statistics, by adapting randomized response to handle a larger domain of possibilities [9, 28, 57, 67]. The current state of the art solutions involve a combination of ideas from data transformation, sketching and hash projections to reduce the communication cost for each user, and computational effort for the data aggregator to put the information together [57, 67]. Building on this, there has been substantial effort to solve a variety of problems in the local model in the last 5 years.

3.1.1 Large-scale Deployments

Google [29, 68]. Erlingsson *et al.* in their pioneering work [68] presented the RAPPOR system deployed in the Chrome browser to facilitate the collection of browser statistics such as homepage settings, running processes in order to understand user experience and perform malware detection. The core idea of RAPPOR hinges on perturbing each individual's information encoded in a bloom filter [69] via 1 bit randomized response. Fanti *et al.* in [29] extended RAPPOR capabilities to collection of joint distributions between various categories by providing an expectation maximization based decoding mechanism. Their complete solution also enables the estimation of values which are not part of the initial dictionary.

Longitudinal Privacy: The problem of differentially private aggregation of data gathered over time was studied by by Dwork *et al.* [70] in the centralized case. Since continual observations are proven to be vulnerable to various attacks (e.g. [71]), longitudinal privacy was among the main goals of these systems. RAPPOR proposed a heuristical memoization scheme that asks users to memorize their noisy answers and repeat them in response to the same queries about a data value. To prevent identification based on cached answers, RAPPOR adds additional noise.

Apple [28, 72]. One of the shortcomings of Google's deployments was suboptimal communication complexity and inefficient use of privacy budget. Apple's DP implementation was announced in 2016, and is documented in a patent application [72] and subsequent blog post [28]. Similar to Google's usecase, they wanted Apple and app developers to collect usage and typing history to train language models by aggregating frequencies of the most frequently used words/emojies. Their techniques fixed the drawbacks observed in RAPPOR by combining signal processing and data summarization techniques such as count-sketch [73] and count-min sketch [74] to reduce the dimensionality of the massive domain.

Microsoft [9]. Ding *et al.*'s work introduced a histogram primitive and a randomized rounding based discretization scheme to collect data over time from fixed users. A more general theoretical framework to estimate frequencies and heavy hitters under continual observation has been proposed recently by Joseph *et al.* [75] and Erlingsson *et al.* [76].

Snap [12]. Pihur *et al.*considered the problem of building a massively distributed system for training *Generalized Linear Models (GLMs)* in an asynchronous/lock-free fashion. In their novel "draw and discard" framework, server maintains k instances of the machine learning model. Upon client request, a randomly sampled instance is sent to be updated. The client updates and perturbs the received model and server replaces a uniformly sampled model with this updated model at its end. This scheme facilitates asynchronous model update and prevents injection of spam models from malicious clients. More interestingly, they prove

that due to "draw and discard" scheme, the expected intra-model variance remains constant ($\frac{k\sigma}{2}$) after infinite updates when made with a zero mean noise distribution with variance σ.

While deployments at such wide scale was a triumph for DP community, some of them also faced criticism for their opaque implementation and privacy guarantees that erode with time. While memoization schemes provide protection against data being directly exposed, this protection deteriorates when answers to correlated data vary non independently. A more rigourous study on longitudinal privacy was performed by Tang *et al.* in [77]. Through static and dynamic code analysis of macOS Sierra, they discovered that the value of ϵ used per day for each user is as large of 16, which is much higher than typically accepted values in academic circles.

3.1.2 Heavy Hitters

Most quantities (emojies, words, home pages etc.) these deployments tried to estimate had a heavy-tailed distribution. Therefore, aggregating the full histogram was a sub-optimal solution. Naturally, the abstract problem of identifying the most frequent k items a.k.a. *heavy hitters* under local setting attracted a lot of attention in parallel. Bassily and Smith in [78] provided an initial theoretical solution based on random projection matrix. These ideas were generalized to approximate DP (definition 9) by Bun *et al.* [79]. However, these solutions provided acceptable practical results only when the domain from which the items are drawn is significantly larger than the population size. On practical fronts, [80, 81] proposed simple interactive histogram frequency oracle based protocols in the case when each users holds a set of items of unequal length. Sketching and dimensionality reduction techniques from Apple's work were refined and extended by Bassily *et al.* [67]. Their methods match the state of art theoretical bounds on accuracy and offer various resource trade-offs. More recently, work by Jia and Gong [82] propose a post-processing scheme that incorporates prior knowledge about noise mechanism and the true frequencies and improves the estimates of any heavy hitter algorithm.

3.1.3 Social Networks

Much sensitive individual data is best represented as a graph—either a simple graph between users, or a bipartite graph between users and other entities. Recent work has aimed at building accurate synthetic graph models under more relaxed *edge* LDP [83] that applies to adjacency lists differing by an edge. In their multiround protocol, aggregator randomly clusters users into k groups. Users know the frequency distribution of his neighbors for all groups. Next users perturb their histograms with zero mean Laplace mechanism and send it to the aggregator. In order to preserve structure, the nodes with similar histograms should be clustered in the same group and vice versa. They iteratively improve the the quality of their

clustering through multiple rounds of data collection. Once a good clustering is achieved, a random graph generation model is used to assign inter/intra cluster edges. Novelty of this work lies in finding the group size k that minimizes the error between true and perturbed degree vectors.

3.1.4 Location Data

Service providers may be interested in estimating user density over a spatial domain without learning the users' true location. It's easy to solve this problem using frequency oracles mentioned in Section 2.4.2. However, this solution does not account for varying spatial density. Moreover, users may have varying preferences over the granularity of data collection. For example, some users would not mind sharing their true city but not their block number while others may not be comfortable in providing any detail outside their state. Initial work on this problem has extended LDP private frequency collection [84]. They impose a semantic hierarchy over locations at the level of block, city, state and country. Users specify their location at their choice of granularity and the data is collected using frequency oracles. It is open to extend this to build more sophisticated user movement models.

3.1.5 Machine Learning

- **Supervised Learning.** A large body of research has focused on performing *Empirical Risk Minimization* (ERM) via *stochastic gradient descent (SGD)* for various loss functions including linear/logistic regression and support vector machines. One line of research including [85–87] approached this general problem by designing more accurate perturbation primitives for mean estimation for numeric data since the worst case accuracy of gradient estimation depends mainly on the accuracy of such primitive used. Shin *et al.* [88] used locally differentially private version of SGD proposed in [85] to recommend unrated movies using matrix factorization while preserving privacy of each user's items and ratings. A similar matrix factorization based approach has been developed in the context of crowd-sourcing platforms by [89] to predict the answers to unanswered tasks.

  On the other hand, more theoretical works studied the impact of interactive data collection on accuracy [90] and came up with single round protocols [91–94] using techniques from approximation theory under smoothness/bounded data assumptions.

- **Unsupervised Learning.**

  Clustering. Nissim *et al.* [95, 96] studied the problem of finding a minimum enclosing ball under DP. In this problem, given a set of points N points in $\mathbb{R}^D$, the task is to find the ball of smallest radius with at least $t \leq N$ points. They provide local (and

central) algorithms achieving constant factor approximation and apply them to solve the problem of k-mean clustering.

Text and language modeling. Discovering frequently used words is one of the basic problems in language modeling and was among the main goals of Google's and Apple's deployment. The full details of Apple's algorithm are unknown. RAPPOR finds the frequency of heavy hitter N-grams and then uses a clique finding algorithm to reconstruct a term from each clique. This algorithm provides unsatisfactory accuracy due to overwhelming noise. Wang *et al.* [97] solved this problem by interactively constructing the *trie* datastructure using familiar techniques used in private heavy hitters identification problem.

Crowdsourcing. Platforms like Amazon Mechanical Turk (AMT) release tasks that are easy for humans but remain difficult for computers. The workers complete these tasks in exchange of a reward. Since tasks are completed at varying quality and often most workers only provide answers to a very small fraction of the tasks, developing methods to estimate worker quality (truth inference) and inferring the answers of uncompleted tasks is a key challenge. Sun *et al.* [89] came up with a LDP version of an existing expectation maximization based performing truth inference method to LDP. Their solution also extended the randomized response method to handle the missing values.

3.1.6 Shuffle Model

The large amount of noise required in the local model has motivated the development of other models. For example, the Encode, Shuffle, Analyze (ESA) model introduced by Bittau [98] involves a trusted shuffler that strips all the identifiers and permutes the user messages before sending them to an untrusted aggregator. This model has been brought recently in the local setting by various theoretical works [76, 99, 100] that develop mean and histogram primitives and build a novel theory of privacy amplification due to the shuffle step. However, feasibility of large scale implementation of a trusted shuffler is still an open problem.

3.1.7 Federated Learning

Distributed optimization with emphasis on protecting client privacy is a longstanding goal pursued by many research communities including cryptography, databases, and machine learning. The framework of federated learning (FL) introduced by McMahan *et al.* [101] facilitates decentralized training of a machine learning algorithm under the coordination of a centralized server in scenarios when data is distributed across multiple decentralized edge devices. The key feature of this approach is that the raw local data samples are not

exchanged with the server but instead, focused updates ready for immediate consumption are used to accomplish the optimization objective at hand. To comprehensively understand the challenges introduced by this framework and recent developments in this space, survey paper compiled by Kairouz *et al.* [102] is recommended. Several recent works including [103–105] couple this approach with DP. On the LDP front, while some of the previously mentioned works such as [12, 85, 88, 97] are solving the learning task by loose federation of clients, satisfying the local DP guarantees under FL framework limits the usecases of FL to massive scale deployments due to large noise amounts. There is a need for a DP model that provides utility guarantees between fully local and fully centralized DP model.

3.1.8 Hybrid Model

Local and central models were always considered incompatible until recently. Avent *et al.* [106] proposed a practical hybrid setting in which a large fraction of users require local privacy and others trust the aggregator and are willing to share their true data. Examples of such type is internal beta testers/early adopters. They show that in such scenarios, one can develop more accurate algorithms by leveraging the fact that a small amount of noise is added into beta users' data. Dubey *et al.* [107] later expressed this problem as a mixture model and quantified the efficacy of the general framework via total error and optimal mixing weight. The version of hybrid model both works adopted provides improvement of up to a constant factor over full local and the centralized model for certain regimes of parameters. While this improvement may not seem like much theoretically, constants matter in real world deployments of DP. Understanding full capacities of this framework is an active area of research.

3.2 Five principles for LDP

We abstract five key principles that we observe recurringly in applied LDP literature. Although each individual idea may seem relatively simple, collectively they provide a complete solution, and their combination yields novel results. In summary, these principles, which are generally applicable to other problems as well, are as follows:
(A) Transform the input. Rather than work with the raw input, have users apply (linear) transformation to the input to align it better with the intended application.
(B) Densify the representation. Since each user's input is typically sparse, use techniques from signal processing to densify it and reduce the communication cost.
(C) Compose transformations. Provided that they are linear, multiple transformations can be composed in sequence to obtain the best properties of each.
(D) Use sampling. When multiple pieces of information are needed, the best results are obtained by sampling which to gather from each user, rather than trying to measure them all.

(E) Apply post-processing. Significant gains in accuracy are possible by post-processing the global estimates, to take advantage of consistency and overlap.

3.3 Prior Work On The Problems Of Interest

Next we will discuss the prior work relevant to marginal, range and count queries.

3.3.1 Marginals Queries

Marginal tables arise in many places throughout data processing. For example, an OLAP datacube is the collection of all possible marginals of a data set. Consequently, there has been much work to release individual marginals or collections of marginals under privacy guarantees. To the best of our knowledge, most of these assume the trusted aggregator model. The motivations for these algorithms — accurate statistics collection, data analysis, model building etc. — are just as compelling under the model of LDP which removes the trusted aggregator. We discuss a representative set of approaches from prior works and check whether they can be applied under LDP.

Laplace Noise. The baseline for differential privacy is the sensitivity and noise approach: we bound (over all possible inputs) the "sensitivity" of a target query in terms of the amount by which the output can vary as a function of the input. Adding noise from an appropriate distribution (typically Laplace) calibrated by the sensitivity guarantees privacy. This approach transfers to LDP fairly smoothly, since the sensitivity of a single marginal on N users is easy to bound by $O(1/N)$ [50]. A variant is to apply this to a transformation of the data, such as a wavelet or Fourier transform [108, 109]. Our contribution is to refine and analyze how to release marginals via transformations under the related guarantee of LDP.

Subset Marginal Selection. When the objective is to release many marginals — say, the entire data cube — the above approach shows its limitations, since the sensitivity, and hence the scale of the noise grows exponentially with the number of dimensions: 2^d. Ding *et al.* [110] compute low dimensional marginals by aggregating high dimensional marginals, chosen via a constrained optimization problem and a greedy approximation. This solution does not translate naturally to LDP, since each user has access to only her record and may come up with a different subset locally compared to others.

Multiplicative Weights. Several approaches use the *multiplicative weight update method* to iteratively pick an output distribution [111–113]. For concreteness, we describe a non-adaptive approach due to Hardt *et al.* [111]. The method initializes a candidate output uniform marginal, and repeatedly modifies it so that it is a better fit for the data. To ensure DP, it uses the exponential mechanism [114] to sample a k-way marginal whose projection at a certain point in the true data is far from the corresponding value for the candidate.

The candidate is then scaled multiplicatively to reduce the discrepancy. The sampling and re-scaling step is repeated multiple times, and the convergence properties are analyzed. The number of steps must be limited, as the "privacy budget" must be spread out over all steps to give an overall privacy guarantee. Applying the exponential mechanism in this way does not obviously extend to the LDP model. In particular, every user's single input is almost equally far from any candidate distribution, so it is hard to coordinate the sampling to ensure that the process converges. A natural implementation would have many rounds of communication, whereas we focus on solutions where each user generates a single output without further coordination.

Expectation Maximization. While materialization of marginals has not been the primary focus of prior work, a work due to Fanti *et al.* does suggest an alternative approach for the 2-way marginal case [29] as a part of their solution. The central idea is for each user to materialize information on all d attributes, and to use an iterative post-processing method on the observed combinations of reported values to reach an estimate for a given marginal. We present this idea in more detail and implement it in Section 4.3.4.

In summary, the problem of understanding correlations between a small subset of variables has not been studied carefully under LDP settings by prior work and deserves a fresh study.

3.3.2 Range Queries

Exact range queries can be answered by simply scanning the data and counting the number of tuples that fall within the range; faster answers are possible by pre-processing, such as sorting the data (for one-dimensional ranges). Multi-dimensional range queries are addressed by geometric data structures such as k-d trees or quadtrees [45]. As the dimension increases, these methods suffer from the "curse of dimensionality", and it is usually faster to simply scan the data.

Various approaches exist to approximately answer range queries. A random sample of the data allows the answer on the sample to be extrapolated; to give an answer with an additive ϵ guarantee requires a sample of size $\mathcal{O}(\frac{1}{\epsilon^2})$ [115]. Other data structures, based on histograms or streaming data sketches can answer one-dimensional range queries with the same accuracy guarantee and with a space cost of $\mathcal{O}(1/\epsilon)$ [115]. However, these methods do not naturally translate to the private setting, since they retain information about a subset of the input tuples exactly, which tends to conflict with formal statistical privacy guarantees.

Private Range queries. In the centralized DP model, there has been extensive consideration of range queries. Part of our contribution is to show how some of these ideas can be translated to the local model, and to provide customized analysis for the resulting algorithms. Much

early work on DP histograms considered range queries as a natural target [49, 116]. However, simply summing up histogram entries leads to large errors for long range queries.

Xiao *et al.* [117] considered adding noise in the Haar wavelet domain, while Hay *et al.* [118] formalized the approach of keeping a hierarchical representation of data. Both approaches promise error that scales only logarithmically with the length of the range. These results were refined by Qardaji *et al.* [119], who compared the two approaches and optimized parameter settings. The conclusion there was that a hierarchical approach with moderate fan-out (of 16) was preferable, more than halving the (squared) error from the Haar approach. A parallel line of work considered two-dimensional range queries, introducing the notion of private spatial decompositions based on k-d trees and quadtrees [120]. Subsequent work argued that shallow hierarchical structures were often preferable, with only a few levels of refinement [121].

While the range queries problem has been subject of interest for many works in the centralized model, it has not been addressed at all in the local setting by any work.

3.3.3 Count Queries

The most relevant work to our interests is due to Ghosh *et al.* [36] who studied the problem of designing mechanisms optimizing for expected utility. Their contributions are to introduce a linear programming formulation of the problem, and to show that a certain mechanism (denoted GM) emerges as the basis of other optimal mechanisms. Gupte and Sundararajan proved a similar universality result for *minimax* loss functions and uniform weights w_j [37]. They provided a simple test for when a given mechanism can be obtained by first applying GM and then modifying the result (e.g. by randomly sampling from a distribution indexed by the observed output from GM). Subsequent work by Brenner and Nissim [122] shows that such *universally optimal* mechanisms are not possible in general for other computations, such as computing histograms.

In this thesis, we limit our attention to Ghosh *et al.*'s expected utility model and begin our study in chapter 6 by observing some anomalies in their framework when employed in practice.

Chapter 4

Marginal Queries

4.1 Chapter Outline And Our Contributions

In this chapter (based on [40]), we provide a general framework for marginal release under LDP, with theoretical and empirical analysis.

• We first recall the setting in which we want to answer the marginal queries (Section 4.2). We then describe a set of new algorithms that give unbiased estimators for marginals, which vary on fundamental design choices such as whether to release information about each marginal in turn, or about the whole joint distribution; and whether to release statistics directly about the tables, or to give derived statistics based on (Fourier) transforms of the data. For each combination, we argue that it meets the LDP guarantee, and provide an accuracy guarantee in terms of the privacy parameter ϵ, population size N, and also the dimensionality of the data, d, and target marginals, k (Section 4.3).

• We perform experimental comparison to augment the theoretical understanding in Section 4.4, focusing mostly on the low-degree marginals that are of most value.

• Across a range of data dimensionalities and marginal sizes, the most effective techniques are based on working in the Fourier (Hadamard) transform space, which capture more information per user than methods working directly in the data space. The use of Hadamard transform for materializing marginals was considered by early work in the centralized differential privacy model, but has fallen from favour in the centralized model, supplanted by more involved privacy mechanisms [110, 111, 123]. We observe that these other mechanisms do not easily translate to the local model.

• Concurrent with the development of the work on which this chapter is based on, the Hadamard basis has found application in protocols for LDP frequency estimation [28]. There, incorporating the transform preserves the accuracy guarantees, while reducing the communication cost. In our setting, we show that the transform can both improve accuracy

and reduce communication cost. The endpoint of our evaluation is the application of our methods to two use-cases: building a Bayesian model of the data, and testing statistical significance of correlations. These confirm that in practice the Hadamard-based approach is preferable and the most scalable in terms of communication and computation cost.

4.2 Model And Preliminaries

In line with prior work [109], our main focus is on data represented by binary variables. This helps to keep the notation uniform, and highlights the key challenges.

In our setting, each user i has a private bit vector $j_i \in \{0,1\}^d$ that represents the values of the d (sensitive) attributes for i. It is often more convenient to view the user's data instead as an indicator vector z_i of length $D = 2^d$ with 1 at exactly one place j_i and 0's at remaining positions. The domain of all such z_i's is the set of identity basis vectors $\mathcal{I}_{2^d \times 2^d}$. This 'unary' view of user data allows us to model the full contingency table correspondingly as a vector (histogram) of length 2^d with each cell indexed by $\eta \in \{0,1\}^d$ storing the count of all individuals with that exact combination of attribute values. This encoding is also called *one hot encoding*.

An untrusted aggregator (e.g. a pollster) is interested in gathering information on these attributes from the population of users. Under the LDP model, the aggregator is not allowed (on legal/ethical grounds) to collect any user i's records in plain form. The gathered data should allow running queries (e.g. the fraction of users that use product A, B but not C together) over the interaction of at most $k \leq d$ attributes. We do not assume that there is a fixed set of queries known a priori. Rather, we allow arbitrary such queries to be posed over the collected data. Our goal is to allow the accurate reconstruction of k-way marginal tables under LDP.

Definition 15 (Marginal release problem)**.** *Given a set of N users, our aim is to collect information (with an LDP guarantee) to allow an approximation of any k-way marginal β of the full d-way distribution $z = \frac{\sum_{i=1}^{N} z_i}{N}$. Let $\widehat{\mathcal{C}^\beta}$ be the approximate answer. We measure the quality of this in terms of the* total variation distance *from the true answer $\mathcal{C}^\beta(z)$, i.e.*

$$\frac{1}{2}\sum_{\gamma \preceq \beta} |\widehat{\mathcal{C}^\beta}[\gamma] - \mathcal{C}^\beta(z)[\gamma]| = \frac{1}{2}\|\widehat{\mathcal{C}^\beta} - \mathcal{C}^\beta(z)\|$$

The marginals of contingency tables allow the study of interesting correlations among attributes. Analysts are often interested in marginals with relatively few attributes (known as low-dimensional marginals). If we are only concerned with interactions of up to at most k attributes, then it suffices to consider the k-way marginals, rather than the full contingency table. Since during the data collection phase we do not know a priori which of the k-way marginals may be of interest, our aggregation should gather enough information

$$\frac{1}{\sqrt{8}}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

Figure 4.1: Hadamard Transform Matrix for $D = 8$.

from each user to evaluate the set of full k-way marginals for some specified k. Our aim is to show that we can guarantee a small total variation distance with at least constant probability[1]. We will express our bounds on this error in terms of the relevant parameters N, d, k, and the privacy parameter ϵ. To facilitate comparison, we give results using the $\tilde{\mathcal{O}}$ notation which suppresses factors logarithmic in these parameters.

Marginals and Basis Transforms. Since the inputs and marginals of individual users are sparse, the information within them is concentrated in a few locations. A useful tool to handle sparsity and "spread out" the information contained in sparse vectors is to transform them to a different orthonormal basis. There are many well-known transformations which offer different properties, e.g Taylor expansions, Fourier Transforms, Wavelets, Chebyshev polynomials, etc. Among these, the discrete Fourier transformation over the Boolean hypercube—known as the Hadamard transform—has many attractive features for our setting.

Definition 16 (Hadamard Transformation (HT))**.** *The transform of vector $z \in \mathbb{R}^{2^d}$ is $\theta = \phi z$ where ϕ is the orthogonal, symmetric $2^d \times 2^d$ matrix with $\phi_{i,j} = 2^{-d/2}(-1)^{\langle i,j \rangle}$.*

Consequently, each row/column in ϕ consists of entries of the form $\pm\frac{1}{2^{d/2}}$, where the sign is determined by the number of 1 bit positions that i, j agree on, denoted as an inner-product $\langle i, j \rangle$. It is straightforward to verify that any pair of rows ϕ_i, ϕ_j satisfy $\langle \phi_i, \phi_j \rangle = 1$ iff $i = j$, and the inner product is 0 otherwise. Hence ϕ is an orthonormal basis for $\mathbb{R}^{2^d}$. Figure 4.1 illustrates the Hadamard matrix of size $D = 8$. Given an arbitrary vector z, we say that its representation under the HT is given by the 2^d *Hadamard coefficients* (denoted as θ) in the vector $\theta = \phi z$. These properties of HT are well-known due to its role in the theory of Boolean functions [124]. In our case when z_i has only a single 1 (say at index ℓ), the Hadamard transform of z_i amounts to selecting the ℓth basis vector of ϕ, and so $\theta_j = \phi_{j,\ell}$. We rely on two elements to apply the Hadamard transform in our setting. The first follows from the fact that the transform is linear:

[1] All our methods allow the probability of larger error to be made arbitrarily small.

Lemma 5. $\phi(\sum_{i=1}^{n} z_i/N) = \frac{1}{N}\sum_{i=1}^{n}(\phi z_i)$

That is, the Hadamard coefficients for the whole population are formed as the sum of the coefficients from each individual. The second ingredient due to Barak *et al.* [109] is that we can write any marginal $\beta \in \mathcal{C}$ as a sum of only a few Hadamard coefficients.

Lemma 6 ([109])**.** *Hadamard coefficients* $H_k = \{\theta_\alpha : |\alpha| \leq k\}$ *are sufficient to evaluate any* k*-way marginal* β*. Specifically,*

$$\mathcal{C}^\beta(t)_\gamma = \sum_{\alpha \preceq \beta} \langle \phi_\alpha, t \rangle \sum_{\eta : \eta \wedge \beta = \gamma} \phi_{\alpha,\eta} = \sum_{\alpha \preceq \beta} \theta_\alpha \Big(\sum_{\eta : \eta \wedge \beta = \gamma} \phi_{\alpha,\eta} \Big) \tag{4.1}$$

Considering Example 2.2.1, to compute the marginal corresponding to $\beta = 0101$, we just need the four Hadamard coefficients indexed as $\theta_{0000}, \theta_{0001}, \theta_{0100}$ and θ_{0101}. Moreover, to evaluate *any* 2-way marginal from $d = 4$, we just need access to the $\binom{4}{0} + \binom{4}{1} + \binom{4}{2} = 11$ coefficients whose indices have at most 2 non-zero bits, out of the $2^4 = 16$ total coeffcients.

4.3 Private Marginal Release

We identify a number of different algorithmic design choices for marginal release under LDP. By considering all combinations of these choices, we reach a collection of six distinct baseline algorithms, which we evaluate analytically and empirically, and identify some clear overall preferred approaches from our subsequent study. We describe our algorithms in terms of two dimensions:

View of the data. The first dimension is to ask what view the algorithm takes of the data. We are interested in marginals, so one approach is to project the data out into the set of marginals of interest, and release statistics about those marginals. However, since any marginal can be obtained from the full input distribution by aggregation, it is also possible to work with the data in this form.

How the information is released. The canonical way to release data under LDP is to apply Randomized Response. As discussed in Section 2.4.2, when the user's data is represented as a sparse input vector, we can perturb all the cells in their table by applying Randomized Response; or by reporting a single cell index (via the preferential sampling approach (INPPS) from Section 2.4.2). The alternative approach we study is to apply the Hadamard transform: the user's table is now represented by a collection of coefficients, each of which can take on one of two possible values. We can then sample one Hadamard coefficient, and report it via randomized response (we call this the HT approach).

4.3.1 Accuracy Guarantees

In order to analyze our algorithms, we make use of bounds from statistical analysis, in particular (simplified forms of) the Bernstein and Hoeffding inequalities:

Definition 17 (Bernstein and Hoeffding inequalities)**.** *Given N independent variables X_i such that $\mathbb{E}[X_i] = 0$, $|X_i| < M_i$, and $\mathrm{Var}[X_i] = \sigma^2$ for all i. Then for any $c > 0$,*

$$\Pr\left[\frac{|\sum_{i=1}^{N} X_i|}{N} > c\right] \leq \begin{cases} 2\exp(-\frac{Nc^2}{2\sigma^2 + \frac{2c}{3}\max_i M_i}) & \textit{(Bernstein inequality)} \\ 2\exp(-\frac{N^2c^2}{2\sum_{i=1}^{N} M_i^2}) & \textit{(Hoeffding inequality)} \end{cases}$$

These two bounds are quite similar, but Bernstein makes greater use of the knowledge of the variable distributions, and leads to stronger bounds for us when we can show $\sigma^2 < M = \max_i M_i$.

Master Theorem for Accuracy. To analyze the quality of the different algorithms, we provide a generalized analysis that can be applied to several of our algorithms in turn. We assume that each user input is in $\{-1, 1\}$ in the proof, but we will also be able to apply the theorem when inputs range over other values.

Theorem 2. *Let each z_i be a sparse vector where one entry is $\{-1, 1\}$, and the rest are zero. When each user i samples an input element j with probability p_s and applies randomized response with p_r to construct z_i^*, for $c > 0$ we have*

$$\Pr\left[\frac{|\sum_{i=1}^{N} z_i^*[j] - z_i[j]|}{N} \geq c\right] \leq 2\exp\Big(-\frac{Nc^2 p_s(2p_r - 1)}{2p_r(2\frac{1-p_r}{2p_r-1} + \frac{c}{3})}\Big)$$

Proof. We first consider the input of a single user subject to randomized response, and obtain an unbiased estimate for their contribution to the population statistics. This lets us combine the estimates from each user to compute an unbiased estimate for the population, whose variance we analyze to bound the overall error.

Let $z_i[j] \in \{-1, 1\}$ be i's unknown true input at location j and $z_i^*[j]$ be the unbiased estimate of $z_i[j]$. First, we derive the values we should ascribe to z^* to ensure unbiasedness, i.e. $\mathbb{E}[z_i^*[j]] = z_i[j]$.

1. When j is sampled (with probability p_s) and $z_i[j] = 1$, we set $z_i^*[j] = x/p_s$ with probability p_r and $z_i^*[j] = y/p_s$ otherwise.

2. When j is sampled (with probability p_s), and $z_i[j] = -1$, we set $z_i^*[j] = y/p_s$ with probability p_r and x/p_s otherwise.

3. When j is not sampled, we implicitly set $z_i^*[j] = 0$.

We can encode these conditions with linear equations:

$$p_r x + (1 - p_r)y = -1 \tag{4.2}$$

$$p_r y + (1 - p_r)x = 1 \tag{4.3}$$

Solving, we obtain $x = \frac{1}{(2p_r - 1)}$ and $y = -\frac{1}{(2p_r - 1)}$. As we require $p_r > \frac{1}{2}$, we have $x > 0$ and $y = -x < 0$. We now analyze the (squared) error from using these parameters. Define a random variable for the observed error as $Y_i[j] = z_i^*[j] - z_i[j]$. Observe that $\mathbb{E}[Y_i[j]]$ is 0, and

$$|Y_i[j]| \leq \frac{1}{p_s}\left(1 + \frac{1}{2p_r - 1}\right) = \frac{2p_r}{p_s(2p_r - 1)} := M.$$

Furthermore, $|Y_i[j]|$ is symmetric whether $z_i[j] = 1$ or -1. Then:

$$\begin{aligned}
&\mathrm{Var}[Y_i[j]] = \mathbb{E}[Y_i^2[j]] \\
&= \frac{p_r p_s}{p_s^2}\Big|\frac{1}{2p_r - 1} - 1\Big|^2 + \frac{(1 - p_r)p_s}{p_s^2}\Big|1 + \frac{1}{2p_r - 1}\Big|^2 + (1 - p_s)1^2 \\
&\leq \frac{p_r}{p_s}\left(\frac{2p_r - 2}{2p_r - 1}\right)^2 + \frac{(1 - p_r)}{p_s}\left(\frac{2p_r}{2p_r - 1}\right)^2 + (1 - p_s) \\
&= \frac{4}{p_s(2p_r - 1)^2}(p_r(1 - p_r)^2 + (1 - p_r)p_r^2) + (1 - p_s) \\
&= \frac{4p_r(1 - p_r)}{p_s(2p_r - 1)^2} + (1 - p_s) := \sigma^2.
\end{aligned} \tag{4.4}$$

Now we consider the effect of aggregating N estimates of the j'th population parameter. Using Bernstein's inequality (Definition 17), we can bound the probability of the error being large based on the bound M on the absolute value of the $Y_i[j]$'s.

$$\begin{aligned}
\Pr\left[\frac{|\sum_{i=1}^N Y_i[j]|}{N} \geq c\right] &\leq 2\exp\left(-\frac{Nc^2}{2\sigma^2 + \frac{2cM}{3}}\right) \\
&\leq 2\exp\left(-\frac{Nc^2}{2(\frac{p_r(1-p_r)}{p_s(2p_r-1)^2} + 1) + \frac{2cp_r}{3p_s(2p_r-1)}}\right) \\
&= 2\exp\left(-\frac{Nc^2}{\frac{2p_r}{p_s(2p_r-1)}(\frac{2(1-p_r)}{(2p_r-1)} + \frac{c}{3}) + 2}\right)
\end{aligned} \tag{4.5}$$

This provides us with the statement of the theorem. □

Intuitively, this theorem lets us express the (total variation) error in a marginal as a function of parameters p_s and p_r. We will choose values of c that make this probability constant — this implies (for example) that c should be chosen proportional to $1/\sqrt{Np_s}$. Hence, we capture how the error decreases as N increases, and how it increases as the number of items being sampled from increases.

4.3.2 Input Perturbation Based Methods

The three approaches which work directly on the input data require a two-step analysis: first we consider the accuracy of reconstruction of some global information (e.g. the full distribution), then we analyze the accuracy of aggregating this to give the required marginal β. Throughout we assume that 2^d is at least $\tilde{\mathcal{O}}(N)$, i.e. the number of users N participating is at least proportional to the number of cells in the full distribution (2^d). This is natural, since it requires our methods which sample cells from the full input to have at least constant probability of probing any given cell. Now we spell out the details of our input perturbation based algorithms. For all of our algorithms, each user i uses the one-hot encoding for her input, so $z_i \in \mathcal{I}_{2^d \times 2^d}$.

Randomized Response On Input (INPRR/OUE). As described in Section 2.4.2, the most direct application of LDP here is to add noise to all 2^d locations.

Perturbation. Each user i perturbs their value z_i at every index $\ell \in 2^d$ using $\frac{\epsilon}{2}$-RR to get $z_i^* \in \mathbb{R}^d$ and sends it to the aggregator.

Aggregation. We reconstruct a version of the full input z^* by simply unbiasing and summing all these contributions (and dividing by N); any desired marginal β is obtained by taking $\mathcal{C}^\beta(z^*)$, i.e. computing that marginal of the reconstructed input.

INPRR though simple, does not scale well with d as expected. It is also potentially costly to apply, since each user needs to materialize and communicate 2^d pieces of information. Applying our general analysis allows us to bound the error (total variation distance) in the returned marginal.

Theorem 3. *With constant probability,* INPRR/OUE *achieves ϵ-LDP and guarantees that*

$$\|\mathcal{C}^\beta(z) - \mathcal{C}^\beta(z^*)\|_1 = \tilde{\mathcal{O}}\Big(\frac{2^{(d+k)/2}}{\epsilon\sqrt{N}}\Big)$$

Proof. We first analyze the accuracy with which each entry of the full marginal $t[j]$ is reconstructed, then combine these to obtain the overall result. Consider an arbitrary index $j \in 2^d$, since INPRR is symmetric across all indices. To achieve ϵ-LDP, we set $p_r = \frac{\exp(\epsilon/2)}{1+\exp(\epsilon/2)}$, and $p_s = 1$. For the purpose of analysis only, we reduce the problem so that we can apply Theorem 2, by applying a remapping from $\{0, 1\}$ to $\{-1, 1\}$: we replace $z_i[j]$ with $z_i'[j] = 2z_i[j] - 1$. Observe that the absolute error in reconstructing $z_i'[j]$ is only a constant factor of that in reconstructing $[j]$. Writing $\alpha = \exp(\epsilon/2)$, then we have the variance of the

local errors $Y_i[j] = (z_i[j] - z_i^*[j])$ is (substituting these values of p_r and p_s into (4.4)):

$$\begin{aligned}\mathrm{Var}[Y_i[j]] &\le 4\frac{p_r(1-p_r)}{(2p_r-1)^2} + 1 - 1 = 4\frac{(\frac{1}{1+\alpha})(1-\frac{1}{1+\alpha})}{(\frac{2}{1+\alpha}-1)^2} \\ &= 4\frac{\frac{\alpha}{(1+\alpha)^2}}{(\frac{1-\alpha}{1+\alpha})^2} = \frac{4\alpha}{(1-\alpha)^2} = \frac{4\exp(-\epsilon/2)}{(\exp(\epsilon/2)-1)^2}.\end{aligned}$$

The reconstruction of the full input distribution is $z^* = \sum_{i=1}^N z_i^*/N$. We can make use of the inequalities $\frac{1}{\exp(\epsilon/2)-1} \le \frac{1}{\epsilon}$ and $1 < \exp(\epsilon/2) < 4$ for $0 < \epsilon < 2$ to bound the variance and substitute into (4.5).

$$\Pr[|z_j - z_j^*| > c] \le 2\exp\left(-\frac{Nc^2}{2\cdot(4\frac{8}{\epsilon^2}) + \frac{2\cdot 8c}{3\epsilon}}\right)$$

Setting c to $9N^{-1/2}\frac{1}{\epsilon}\sqrt{\log 2^{d+1}/\delta}$ bounds this probability to

$$2\exp\left(-\frac{81\frac{1}{\epsilon^2}\log 2^{d+1}/\delta}{\frac{32}{\epsilon^2} + \frac{16}{3}\frac{9}{\epsilon^2}\sqrt{\frac{2^d\log 2^{d+1}/\delta}{N}}}\right) < 2\exp\Big(-\frac{81\log(2^{d+1}/\delta)}{32+48}\Big) \le \delta/2^d$$

This ensures that this error probability is less than $\delta/2^d$ for any index j. This limits the error in each of the 2^d estimates to being $\tilde{\mathcal{O}}(\frac{1}{\epsilon}\sqrt{\frac{1}{N}})$, by applying a union bound.

We construct the target marginal β via the marginal operator, so $\widehat{\mathcal{C}^\beta} = \mathcal{C}^\beta(z^*)$. Each entry $z^*[j]$ is an unbiased estimator for $t[j]$ whose absolute value is bounded by c with probability $1-\delta$. Conditioning on this event, we compute $\widehat{\mathcal{C}^\beta}[\gamma] = \sum_{\alpha\preceq\gamma} z^*[\alpha]$, summing over the 2^{d-k} values of $\alpha \preceq \gamma$. The error in this quantity is then at most $\tilde{\mathcal{O}}(c\sqrt{2^{d-k}})$, applying a Hoeffding bound (Definition 17). Finally, summing the absolute errors over all 2^k entries γ in the target marginal β, we have probability at least $1-\delta$ that the total variation distance is $\tilde{\mathcal{O}}(\frac{2^k 2^{(d-k)/2}}{\epsilon\sqrt{N}}) = \tilde{\mathcal{O}}(\frac{2^{(d+k)/2}}{\epsilon\sqrt{N}})$. □

Preferential Sampling On Input/Generalized Randomized Response (INPPS/GRR). Our second method uses preferential sampling described in Section 2.4.2 to report a (noisy) index, so sends d bits.

Perturbation. Each user i samples the input signal index j with probability p_s, then reports the selected index to the aggregator.

Aggregation. The reconstructed distribution z^* is found by applying the unbiasing to each noisy report , and computing the average. As in the previous case, we can obtain any desired marginal by aggregating the reconstructed distribution. We provide an alternate proof for GRR's estimator variance.

Theorem 4. INPPS/GRR *achieves ϵ-LDP and guarantees that with constant probability we have for a target k-way marginal β*

$$||\mathcal{C}^\beta(t) - \mathcal{C}^\beta(z^*)||_1 = \tilde{\mathcal{O}}\left(\frac{2^{d+k/2}}{\epsilon\sqrt{N}}\right).$$

Proof. Similar to Theorem 2, we define random variables $Y_i[j]$ which describe the error in the estimate from user i at position j. The proof is a bit more complicated here, since these variables are not symmetric. Consider user i who samples a location under INPPS, such that the correct location is sampled with probability p_s, and each of the $D = 2^d - 1$ incorrect locations is sampled with probability $(1 - p_s)/D$. Following the analysis in Section 2.4.2, we report $\frac{D+p_s-1}{Dp_s+p_s-1}$ for the location which is sampled, and $\frac{p_s-1}{Dp_s+p_s-1}$ for those which are not sampled. For convenience, define the quantity $\Delta = Dp_s + p_s - 1$. The choice of p_s (which depends on D and ϵ) ensures that $\Delta > 0$. There are two cases that arise:

(i) $z_i[j] = 1$. With probability p_s, location j is sampled. The contribution to the error at this location is $\frac{D+p_s-1}{\Delta} - 1 = \frac{1}{\Delta}(D + p_s - 1 - Dp_s - p_s + 1) = \frac{D}{\Delta}(1 - p_s)$.
Else, with probability $1-p_s$, j is not sampled, generating error $\frac{p_s-1}{\Delta}-1 = \frac{p_s-1-Dp_s-p_s+1}{\Delta} = \frac{D}{\Delta}p_s$ for $|z_i^*[j] = z_i[j]|$.
(ii) $z_i[j] = 0$. With probability $\frac{1-p_s}{D}$, we sample this j, giving error $\frac{D+p_s-1}{\Delta} - 0$. Otherwise, the contribution to the error is $\frac{p_s-1}{\Delta}$.

We define a random variable $Y_i[j]$, which is the error resulting from user i in their estimate of $z_i[j]$. Note that an upper bound M on $Y_i[j]$ is D/Δ. We compute bounds on Y_i^2, conditioned on $z_i[j]$.

$$\begin{aligned}
\mathbb{E}[Y_i[j]^2|z_i[j] =1] &= p_s\Big(\frac{D}{\Delta}(1-p_s)\Big)^2 + (1-p_s)\Big(p_s\frac{D}{\Delta}\Big)^2 \\
&= p_s(1-p_s)\Big(\frac{D}{\Delta}\Big)^2 \leq (1-p_s)\frac{D^2}{\Delta^2} \\
\mathbb{E}[Y_i[j]^2|z_i[j] =0] &= \frac{1-p_s}{D}\left(\frac{D+p_s-1}{\Delta}\right)^2 + \left(1-\frac{1-p_s}{D}\right)\left(\frac{p_s-1}{\Delta}\right)^2 \\
&= \frac{1-p_s}{\Delta^2}\left(\frac{1}{D}(D+p_s-1)^2 + \frac{D+p_s-1}{D}(1-p_s)\right) \\
&= \frac{1-p_s}{D\Delta^2}(D+p_s-1)(D+p_s-1+1-p_s) \\
&= (1-p_s)(D+p_s-1)/\Delta^2 \leq (1-p_s)D/\Delta^2
\end{aligned}$$

To bound the error in $z^*[j]$, we make use of the (unknown) parameter f_j, the proportion of users for whom $z_i[j] = 1$. We subsequently remove the dependence on this

quantity. We now write

$$\mathbb{E}[Y_i[j]^2] \leq (1-p_s)\frac{D}{\Delta^2}(f_j D + (1-f_j)) := \sigma_j^2$$

Using this in the Bernstein inequality (Definition 17), we obtain

$$\begin{aligned}\Pr\left[\frac{|\sum_{i=1}^N Y_i[j]|}{N} \geq c_j\right] &\leq 2\exp\left(-Nc_j^2/\left(2\sigma_j^2 + \tfrac{2c_j M}{3}\right)\right) \\ &= 2\exp\left(-\frac{Nc_j^2}{2(1-p_s)\frac{D}{\Delta^2}(f_j D + (1-f_j)) + \frac{2c_j D}{3\Delta}}\right)\end{aligned}$$

If we write $\Psi_j = \sqrt{f_j D + 1 - f_j}$, then setting $c_j = \frac{\sqrt{3D\ln(2/\delta)}}{\Delta\sqrt{N}}\Psi_j$ is sufficient to ensure that this probability is at most δ.When we apply the marginal operator $\mathcal{C}^\beta$ to the reconstructed input z^*, each of the 2^k entries is formed by summing up $(D+1)/2^k$ (unbiased) entries of z^*. Write $f'_\gamma = \sum_{j\wedge\beta=\gamma} f_j$, and define Ψ'_γ correspondingly as $\sqrt{f'_\gamma(D-1) + \frac{D+1}{2^k}}$. Applying the Hoeffding bound (Definition 17), we obtain that each $\mathcal{C}^\beta(z^*)[\gamma]$ has error at most $\frac{\sqrt{3D\ln(2/\delta)}}{\Delta\sqrt{N}}\Psi'_\gamma$ with probability at least $1-\delta$.

We can now sum the error across all $(D+1)/2^k$ indices γ. First,

$$\begin{aligned}\sum_{\gamma\preceq\beta}\psi'_\gamma &= \sum_{\gamma\preceq\beta}(f'_\gamma(D-1) + \frac{D+1}{2^k})^{\frac{1}{2}} \\ &\leq \sqrt{2^k}\Big(\sum_{\gamma\preceq\beta} f'_\gamma(D-1) + \frac{D+1}{2^k}\Big)^{\frac{1}{2}} = \sqrt{2^{k+1}\cdot D}\end{aligned}$$

where the inequality is due to Cauchy-Schwarz, and we use that the f'_γs are a probability distribution, and sum to 1. Then we have a bound on the total variational error error of marginal construction by summing over all indices γ as

$$\sum_{\gamma\preceq\beta}\frac{c'_\gamma}{2} = \frac{1}{2\Delta}\sqrt{\frac{D}{N}}\sqrt{3\ln 2/\delta}\sum_{\gamma\preceq\beta}\Psi'_\gamma \leq \frac{2^{k/2}D}{\Delta\sqrt{N}}\sqrt{\tfrac{3}{2}\ln 2/\delta}$$

We next simplify the term D/Δ as follows. Recall that theory sets $p_s = (1 + D\exp(-\epsilon))^{-1}$. Then

$$\begin{aligned}\frac{D}{\Delta} &= \frac{D}{(D+1)/(1+D\exp(-\epsilon)) - 1} = \frac{D(1+D\exp(-\epsilon))}{D+1-1-D\exp(-\epsilon)} \\ &= \frac{1+D\exp(-\epsilon)}{1-\exp(-\epsilon)} = \frac{1}{1-\exp(-\epsilon)} + \frac{D}{\exp(\epsilon)-1}\end{aligned}$$

When D is very small, in particular when $D=1$, this reduces to a similar error

Algorithm 1 User's routine for INPHT

procedure INPHT(z_i)
  Let $j_i \in \{0,1\}^d$ be the signal index of $z_i \in \mathcal{I}_{2^d \times 2^d}$.
  Randomly sample a coefficient index $\ell_i \in H_k$.
  $\hat{\theta}_i \leftarrow RR(-1^{\langle j_i, \ell_i \rangle})$ ▷ Randomized response on (scaled) θ_{ℓ_i}
  Send $(\hat{\theta}_i, \ell_i)$

Algorithm 2 Aggregator's routine for INPHT

$\Theta^*[0] = 1$ ▷ 0th Hadamard coefficient is always 1.
Aggregator fills table H from tuples $(\hat{\theta}_i, \ell_i)$ as $H_i[\ell_i] = \hat{\theta} i$.
for all $j \in T$ **do**
  $\Theta^*[j] \leftarrow 2^{-d} \frac{\sum_{i=1}^{N} H_i^*[j]}{N_j(2p-1)}$. ▷ N_j is the frequency count of index j.

as for the RR case. Assuming that ϵ is at most a constant (say, 8), we can upper bound this expression by $\mathcal{O}(\frac{D+1}{\epsilon})$. Hence, the total variational error is bounded by $\tilde{\mathcal{O}}(\frac{2^{k/2}(D+1)}{\epsilon\sqrt{N}})$. □

Consequently, we get a guarantee for INPPS/GRR in terms of total variation distance of $\tilde{\mathcal{O}}\left(\frac{2^{k/2}2^d}{\epsilon\sqrt{N}}\right)$. This exceeds the bound of the previous algorithm by a factor of $2^{d/2}$, so we expect the former to be more accurate in practice.

Random Sampling Over Hadamard Coefficients (INPHT). In this method, user i takes the HT of her input and perturbs a uniformly sampled coefficient and releases it via Randomized Response. According to Lemma 6, we do not need to sample from all coefficients; we need only the set of coefficients T sufficient to reconstruct the k-way marginals. T consists of those coefficients whose d-bit (binary) indices contain at most k 1's. There are $|T| = \sum_{\ell=1}^{k} \binom{d}{\ell} = \mathcal{O}(d^k)$ of these, which can be much smaller than the 2^d parameters needed to describe the full input.

Perturbation. Each i samples a coefficient index $\ell_i \in T$ uniformly and computes a scaled-up version of the ℓ_ith Hadamard coefficient θ_i as $\theta_i = (-1)^{\langle j_i, \ell_i \rangle}$. She then perturbs θ_i with ϵ-RR as $\hat{\theta}_i$ and releases the tuple $(\ell_i, \hat{\theta}_i)$.

Aggregation. The aggregator then unbiases, averages and re-scales each noisy coefficient θ_j to estimate $\hat{\theta}_j$. These can then be used to reconstruct any target marginal β via the application of Lemma 6 to generate $\mathcal{C}^\beta(z^*)$.

For completeness, Algorithms 1 and 2 spell out the transformation steps followed by user and aggregator in INPHT. Note that the communication per user can be encoded using 1 bit to describe the output of RR $\hat{\theta}_i$, plus at most d bits to specify ℓ_i, the sampled coefficient. We apply Theorem 2 to this setting to bound the total variation distance between true and reconstructed marginals.

Theorem 5. INPHT *achieves ϵ-LDP, and with constant probability we have for any target k-way marginal β*

$$\|\mathcal{C}^\beta(z) - \mathcal{C}^\beta(z^*)\|_1 = \tilde{\mathcal{O}}\left(\frac{(2d)^{k/2}}{\epsilon\sqrt{N}}\right)$$

Proof. The proof proceeds along the same lines as for Theorem 3. We set $p_r = \exp(\epsilon)/(1+\exp(\epsilon))$ to ensure that INPHT meets ϵ-LDP. Recall that, from Lemma 5, our aim is to compute Hadamard coefficients as the normalized sum of the coefficients from each user. To apply the Master theorem (Theorem 2), we first multiply up each coefficient by the $2^{d/2}$ factor from the Hadamard coefficients θ (Definition 16). Since each user's input vector has only a single 1 entry, this ensures that each $\theta_i[j]$ is either -1 or $+1$. Now the θ_i and θ_i^*s represent the T necessary and sufficient (scaled up) Hadamard coefficients, and so we set $p_s = 1/T$. We write the variance of the errors in these estimates $Y_i[j]$, and obtain

$$\mathrm{Var}[Y_i[j]] = 4T\frac{p_r(1-p_r)}{(2p_r-1)^2} + 1 = \frac{4Te^\epsilon}{(e^\epsilon-1)^2} + 1 = \mathcal{O}(T/\epsilon^2) \tag{4.6}$$

Substituting this variance bound into (4.5), we obtain

$$\Pr\left[\frac{|\sum_{i=1}^N Y_i[j]|}{N} \geq c\right] \leq 2\exp\left(-\frac{Nc^2}{O(T/\epsilon^2 + \frac{Tc}{\epsilon})}\right)$$

Setting c proportional to $N^{-1/2}\frac{1}{\epsilon}\sqrt{T\cdot\log T/\delta}$ ensures that this probability is at most δ/T for any given Hadamard coefficient j (again using that N is large enough). This bound then holds for all T with probability $1-\delta$, using the union bound.

In order to translate this into a bound on the accuracy of reconstructing a marginal, we make use of Lemma 6, that the marginal can be expressed in terms of a linear sum of Hadamard coefficients. Adapting (4.1), we have that

$$\sum_{\gamma\preceq\beta}|\mathcal{C}^\beta[\gamma] - \widehat{\mathcal{C}^\beta}[\gamma]| \leq \sum_{\gamma\preceq\beta}\Big|\sum_{\alpha\preceq\beta}(\theta_\alpha - \hat{\theta}_\alpha)\sum_{\eta\wedge\beta=\gamma}\phi_{\alpha,\eta}\Big|$$

To bound this quantity, we observe that:
(i) There are 2^k such $\gamma\preceq\beta$ to consider.
(ii) There are similarly 2^k such α to consider, and the above analysis bounds $(\theta_\alpha - \hat{\theta}_\alpha) \leq c/2^{d/2}$, once we rescale the coefficients back down. Since the $\hat{\theta}_\alpha$ are unbiased estimators bounded by $c2^{-d/2}$, by the Hoeffding inequality, we have that the sum of 2^k of these is at most $2^{k/2-d/2}c$ with probability at least $1-\delta$.
(iii) Given $\gamma\preceq\beta$, there are 2^{d-k} such η to consider, and so we have $|\sum_{\eta\wedge\beta=\gamma}\phi_{\alpha,\eta}| \leq 2^{d-k}2^{-d/2} = 2^{d/2-k}$.

Then the total variational error is (multiplying these three quantities together) $2^k 2^{k/2-d/2} c 2^{d/2-k} = c2^{k/2} = \tilde{O}\left(\frac{2^{k/2}\sqrt{T}}{\epsilon\sqrt{N}}\right)$. □

Comparing this to the previous results, we observe that the dependence on $2^{k/2}/(\epsilon\sqrt{N})$ is the same. However, our full analysis shows a dependence on $\sqrt{T}$ in place of $\sqrt{2^d}$. Recall that $T = \sum_{\ell=1}^{k} \binom{d}{k} < 2^d$ for $k < d$. For small values of k, this is much improved. For example, for $k = 2$, $\sqrt{T} < d$ in INPHT compared to a $2^{d/2}$ term for INPRR.

4.3.3 Marginal Perturbation Based Methods

Our next methods are the analogs of the Input perturbation methods, applied to a randomly sampled marginal rather than the full input. We only provide the sketches of these proofs to avoid repetition since they are adaptations of the previous proofs.

INPRR/OUE On A Random Marginal (MARGRR). In MARGRR, user i materializes a random marginal $\beta_i \in \mathcal{C}$, then perturbs it using INPRR.

Perturbation. User i samples a random marginal $\beta_i \in \mathcal{C}$, and evaluates its 2^k indices ($\mathcal{C}^\beta(z_i)$) on her input. Note that $\mathcal{C}^\beta(z_i)$ is also sparse. The user then perturbs each index of $\mathcal{C}^\beta(z_i)$ with $\frac{\epsilon}{2}$-RR (INPRR) and sends the tuple $\langle \mathcal{C}^\beta(z_i^*), \beta_i \rangle$ to the aggregator.

Aggregation. The aggregator sums up the perturbed marginals received from all users and unbiases them.

Analysis (outline). As with INPRR, it is immediate that the method achieves ϵ-LDP, since each perturbed marginal index is specific to the input, and is obtained via RR which is ϵ-LDP. We require at most d bits to identify which marginal was chosen, plus 2^k bits to encode the user's perturbed marginal. In terms of accuracy, the analysis is also very similar to INPRR. The difference is that we are now considering sampling from $\binom{d}{k}$ marginals, each of which contains 2^k pieces of information. So where before we had a dependence on 2^d, the method now also depends on $1/p_s = \binom{d}{k} = \mathcal{O}(d^k)$. Thus, via Theorem 3, we obtain a bound on the error in each entry of each marginal of $\tilde{\mathcal{O}}(\sqrt{d^k}/\epsilon\sqrt{N})$. Summing this over the 2^k entries in the marginal, we obtain a total error of $\tilde{O}\left(\frac{2^k d^{k/2}}{\epsilon\sqrt{N}}\right)$.

INPPS/GRR On A Random Marginal (MARGPS). As an alternative approach to MARGRR, we can use preferential sampling (Section 2.4.2) to perturb the sampled marginal. We can pick the entry in the randomly sampled marginal which contains the 1 and apply preferential sampling on it. For small marginals (i.e. small k), this may be effective. Otherwise the algorithm is similar to MARGRR, and we build all the required marginals by averaging together the (unbiased) reported results from all participants.

Analysis. The behaviour of this algorithm can be understood by adapting the analysis of INPPS. Since we work directly with the marginal of size k, we now obtain a bound in terms of $2^{3k/2}$ where before we had $2^{d+k/2}$. However, the effective population size is split

uniformly across the $\binom{d}{k}$ different marginals. Consequently, the total variation distance is $\tilde{\mathcal{O}}\left(\frac{2^{3k/2}d^{k/2}}{\epsilon\sqrt{N}}\right)$. This exceeds the previous result by a factor of $2^{k/2}$, but for small k (such as $k = 2$ or $k = 3$), this can be treated as a constant and the other factors hidden in the big-Oh notation may determine the true behaviour. The user sends d bits to identify the sampled marginal, plus k bits to identify the sampled index within it.

Hadamard Transform Of A Random Marginal (MARGHT). MARGHT also deviates from MARGRR only in how the chosen marginal is materialized: it takes the Hadamard transform of each user's sampled marginal, and uses RR to release information about a randomly chosen coefficient. These are aggregated to obtain estimates of the (full) transform for each k-way marginal β. Note that this method does not share information between marginals, and so does not obtain as strong a result as INPHT.

Analysis. Here, p_r is the same as in INPHT, but we are now sampling over a larger set of possible coefficients: each marginal requires 2^k coefficients to materialize, and we sample across $T = O(d^k)$ marginals. This sets $p_s = \mathcal{O}((2d)^{-k})$. We obtain that $\sigma^2 = \mathcal{O}((2d)^k/\epsilon^2)$ and $M = \mathcal{O}((2d)^k/\epsilon)$. Thus, we bound the absolute error in each reconstructed coefficient by $\tilde{\mathcal{O}}\left(\frac{d^{k/2}}{\epsilon\sqrt{N}}\right)$, by invoking Theorem 2 with these values and then applying the rescaling by 2^{-k}. We directly combine the 2^k coefficients needed by marginal β, giving total error $\tilde{\mathcal{O}}(\frac{2^{3k/2}d^{k/2}}{\epsilon\sqrt{N}})$, similar to the previous case. The communication cost is d bits to identify the marginal, and $k + 1$ bits for the index of the Hadamard coefficient and its perturbed value.

Summary of marginal release methods. Although different in form, all three marginal based methods achieve similar asymptotic error, which we state formally as follows:

Lemma 7. *Two marginal-based methods (*MARGPS *and* MARGHT*) achieve ϵ-LDP and with constant probability the total variation distance between true and reconstructed k-way marginals is at most $\tilde{\mathcal{O}}(\frac{2^{3k/2}d^{k/2}}{\epsilon\sqrt{N}})$. For* MARGRR*, the bound is $\tilde{\mathcal{O}}(\frac{2^k d^{k/2}}{\epsilon\sqrt{N}})$.*

Comparison of all methods. Comparing all six methods, a dependence on a factor of $\frac{2^{k/2}}{\epsilon\sqrt{N}}$ is common to all. Marginal-based methods multiply this by a factor of at least $(2d)^{k/2}$, while input based methods which directly materialize the full marginal (INPRR and INPPS) have a factor of 2^d. The input Hadamard approach INPHT reduces this to just $d^{k/2}$. Asymptotically, we expect INPHT to have the best performance. However, for the parameter regimes we are interested in (e.g. $k = 2$), all these bounds could be close in practice. Hence, we evaluate the methods empirically to augment these bounds. The time cost of all methods is linear in the size of the communication: each user's time cost is proportional to the size of the message sent, while the aggregator's time is proportional to the total size of all messages received, to simply sum up derived quantities. Table 4.1 summarizes these bounds, showing the communication cost (in bits), along with the leading error behaviour (supressing logarithmic factors and the common factor of $\epsilon/\sqrt{N}$).

Method	Communication cost	Error behaviour
INPRR/OUE	2^d	$2^{k/2}2^d$
INPPS/GRR	d	$2^{k/2}2^d$
INPHT	$d+1$	$2^{k/2}d^k$
MARGRR	$d+2^k$	$2^k d^{k/2}$
MARGPS	$d+k$	$2^{3k/2}d^{k/2}$
MARGHT	$d+k+1$	$2^{3k/2}d^{k/2}$

Table 4.1: Summary of communication and error bounds.

N	d	k	ϵ	Failed/Total Marginals
2^{16}	8	1	0.2	3/8
2^{18}	8	2	0.1	15/28
2^{16}	8	2	0.2	3/28
2^{16}	12	2	0.2	19/66
2^{18}	16	2	0.1	120/120
2^{18}	16	2	0.2	72/120
2^{19}	24	2	0.2	276/276

Table 4.2: Failure rate for INPEM on taxi dataset for small ϵ.

4.3.4 Expectation-Maximization (EM) Heuristic

We now discuss the details of Fanti *et al.*'s EM method mentioned briefly in Section 3.3.1. In their scheme, each user independently perturbs each of the d (binary) attributes via (ϵ/d)-randomized response by splitting the budget into d pieces. To reconstruct a target marginal distribution, the aggregator applies an instance of Expectation Maximization (EM). Algorithm 3 describes the EM loop to recover a 2-way marginal. Starting from an initial guess (typically, the uniform marginal), the aggregator updates the guess in a sequence of iterations. Each iteration first computes the posterior distribution given the current guess, applying knowledge of the randomized response mechanism (expectation step). It then marginalizes this posterior using the observed values of combinations of values reported by each user, to obtain an updated guess (maximization step). These steps are repeated until the guess converges, which is then output as the estimated distribution. While this is a plausible heuristic, it does not provide any worst case guarantees of accuracy and more importantly not guaranteed to converge. We compare this method, denoted INPEM, to the algorithms above in our experimental study. In summary, we find that the method provides lower accuracy than our new methods. In particular, we see many examples where it fails: the EM procedure immediately terminates after a single step and outputs the prior (uniform) distribution. Table 4.2 quantifies this in more detail for NYC taxi dataset (to be introduced in Section 4.4.1), and shows some parameter settings where the method fails universally. We

compare INPEM with best of our methods in Section 4.4.4.

Algorithm 3 INPEM Decoding Routine For Constructing 2-way marginal [29].

1: Let β be a 2-way marginal interest and $\gamma \in \{1,0\}^2$ denotes indices of β.
2: $\Pr^0[\gamma] = \frac{1}{2}, \forall \gamma \in \{1,0\}^2$ ▷ Initialize marginal probabilities.
3: $\Omega \in \mathbb{R}$ (e.g. $\Omega = 0.00001$) is a suitable tolerance threshold.
4: **while** $||\Pr^{\tau+1}[\gamma] - \Pr^{\tau}[\gamma]||_\infty \geq \Omega$ **do**
5: $\quad \Pr^{\tau}[\gamma] = \Pr^{\tau+1}[\gamma]$
6: $\quad \Pr^{\tau}[\gamma|\gamma'] = \frac{\Pr^{\tau}[\gamma'|\gamma]\Pr^{\tau}[\gamma]}{\sum_{\gamma \in \{0,1\}^2}\Pr^{\tau}[\gamma'|\gamma]\Pr^{\tau}[\gamma]}, \forall \gamma, \gamma' \in \{1,0\}^2$ ▷ Maximization: Bayes theorem.
7: $\quad \Pr^{\tau+1}[\gamma] = \frac{\sum_{u=1}^{N}\Pr^{\tau}[\gamma|\gamma'_u]}{N}, \forall \gamma, \gamma' \in \{1,0\}^2$ ▷ Expectation: Marginalize the noisy observed distribution.

4.4 Experimental Evaluation

We have two goals for our empirical study: (1) to give experimental confirmation of the accuracy bounds proved above; and (2) to show that our algorithms support interesting machine learning/statistical tasks using our marginal computing machinery as primitives. We implement our methods with standard Python packages (Numpy, Pandas) and perform tests on a standard Linux laptop. Our codebase is publically available [125].

4.4.1 Experimental Setting

Datasets used. We use two sample datasets for our experiments:

NYC Taxi Data [126]. This dataset samples trip records from all trips completed in yellow taxis in NYC from 2013-16. Each trip record can be viewed a unique anonymous rider's response to a set of survey questions about her journey. Some of the attributes are GPS co-ordinates/timestamps of pick-up/drop-off, payment method, trip distance, tip paid, toll paid, total fare etc. From this (very large) data set, we select out the $3M$ records having pickup and/or drop-off locations inside Manhattan. We obtain the 8 binary attributes for each trip listed in Table 2.1. We observe in this dataset that most journeys are short, and so attribute pairs such as pickup/drop-off locations/times, tip-fraction and payment mode are strongly correlated. Meanwhile, most other attribute pairs are negatively correlated, or only weakly related. Since our goal is to privately recover correlation between attributes via marginal distributions, we first confirm its presence the dataset through another well-know metric. The Pearson correlation coefficient [127] measures the linear correlations between a pair of variables. The coefficient ranges from -1 to 1, where 1/-1 is total positive/negative correlation and 0 signifies no linear correlation. Figure 4.2 gives a heatmap for the strength of pairwise associations using the Pearson coefficient.

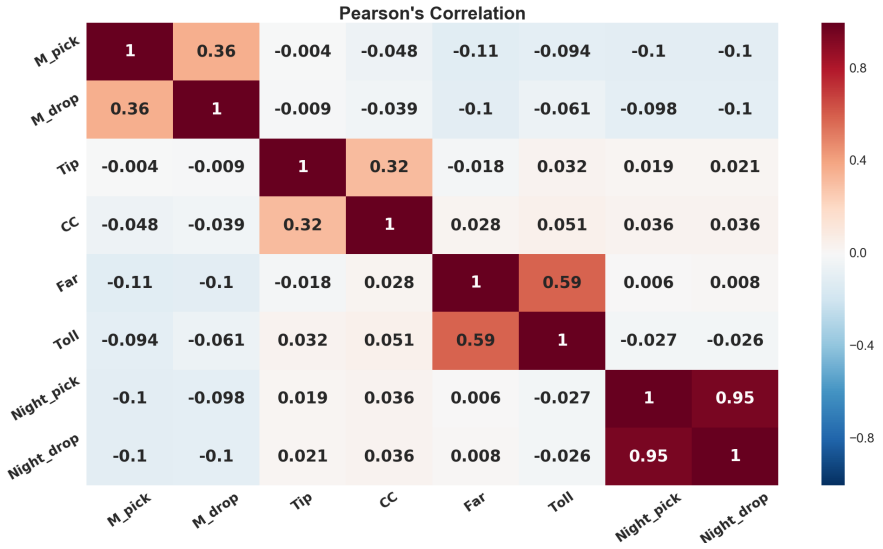


Figure 4.2: Attribute correlation heatmap of NYC taxi dataset.

Movielens [128]. This dataset comprises over $20M$ records from over $150K$ anonymous users who rate nearly $40K$ unique movie titles. Each title belongs to one or more of 17 genres such as Action, Comedy, Crime, Musical etc. From this, we derive a dataset to encode "video viewing" preferences. We first find the top-1000 most rated movies in each genre. We assign each user a vector of preferences $z_i \in \{0,1\}^d$. For each user i, a bit at index $j \in [d]$ is 1 if i has rated at least one of the top 1000 movies of genre j and zero otherwise. In this data, most attribute pairs are postively correlated.

Default Parameters And Settings. In each experimental instance, we uniformly sample (with replacement) a set of random unique records/users ($50K \leq N \leq 0.5M$) as a power of 2 from the total available population. We vary ϵ from 0.2 (higher privacy) to 1.4 (lower privacy). Note that the theory shows that ϵ and N are tightly related: decreasing ϵ means N must be increased to obtain the same accuracy. Some prior work on LDP e.g. [129] studies a smaller regime of ϵ values, at the expense of a much larger user population. We begin our experimental study by sampling (without replacement) a small subset of dimensions d (3-8), and increase to larger dimensionalities for our later experiments. As per our motivation, we focus on small marginals ($k = 1, 2, 3$). We repeat each marginal reconstruction 10 times to observe the consistency in our results, and show error bars.

4.4.2 Impact Of Varying Population Size N

We aim to understand how much a privately reconstructed marginal $\mathcal{C}^\beta(z^*)$ deviates from its non-private counterpart $\mathcal{C}^\beta(t)$ when β is drawn from the set of k-way marginals. First, we

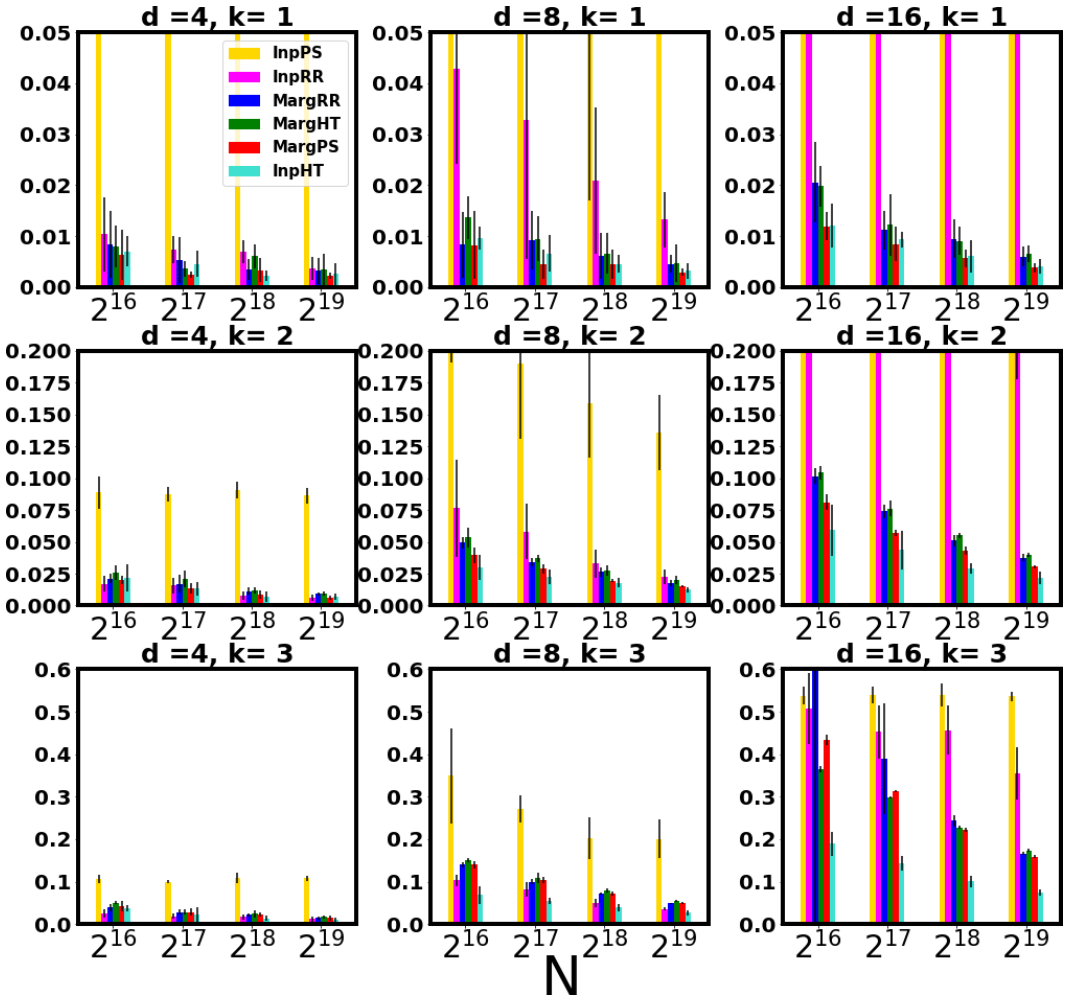


Figure 4.3: Mean total variation distance for $1, 2, 3-$way marginals over the movielens dataset as N varies.

fix $\epsilon = 1.1$ [2] and vary N for different choices of d, k. For our initial comparison, we keep d's moderate ($\{4, 8, 16\}$), as this suffices to distinguish the methods which scale well from those that do not.

Experimental Setting. Figure 4.3 shows plots for total variation distance in reconstruction of k-way marginals as we vary N for all combinations of $k \in \{1, 2, 3\}$ and $d \in \{4, 8, 16\}$ on the movielens dataset with $\epsilon = \ln(3) \approx 1.1$ fixed throughout the experiment. Each grid point shows the mean variational distance of all $k = 1, 2, 3$ marginals. The values of parameters d and k vary across the rows and columns of the figure, respectively.

Experimental Observations. A high level observation across the board is how the error reduces as N increases for all 6 algorithms. This agrees with the analysis that error should be

[2] $\epsilon = 1.1$ is merely a representative value from the privacy parameter ranges that the prior works considered acceptable .

proportional to $1/\sqrt{N}$, i.e. error halves as population quadruples. We also see an increase in error along columns (rows) as k (d) increases, although the dependency varies for different algorithms.

Our second observation is that the performance of INPPS decays rapidly as a function of d, consistent with the accuracy bound of 2^d. Typically, INPPS's error does not reduce as with N. This is because the probability of outputting the signal index becomes so small for larger d's that each user responds with a random index most of the time. This means that the perturbed input distribution does not contain much information for our estimators to invert the added noise with precision. One surrogate for the accuracy of the algorithms is the number of statistics materialized in each case. For $d = 8, k = 2$, INPPS construct $2^8 = 256$ values, while the marginal-based methods are working on $\binom{8}{2} \times 2^k = 112$ values. As a result, the number of data points per cell is proportionately more for MARGHT, MARGPS thus improving their accuracy. On the other hand, the input-based method INPHT convincingly achieves the lowest (or near lowest) error across all parameter settings.

Breaking the algorithms down by the cardinality of the marginal (k), note that for $k = 1$ then the primitives INPRR and INPPS are effectively the same. Further, for a given marginal, there is only one meaningful Hadamard coefficient needed, and so we expect the Hadamard-based methods to behave similarly. Indeed, the methods MARGPS, MARGRR, MARGHT, and INPHT are largely indistinguishable in their accuracy. For the larger 2-way and 3-way marginals, we see more variation in behaviour. The input-based methods do not fare well: INPPS has very large errors for even smaller d values ($d = 4$ and $d = 8$), and INPRR is similar once $d = 16$. We observe that MARGPS achieves better accuracy than MARGRR. This supports the idea that the former method, which preferentially reports the location of each user's input value, can do better than naive randomized response, even though this is not apparent from the asymptotic bounds. Interestingly, on this data we see that the difference in performance of MARGPS and MARGHT is tiny, and MARGPS turns out to be a better algorithm. For $d = 16$, MARGHT starts as a better algorithm but is outperformed by MARGPS.

INPRR is among the better methods for smaller values of d and k's. However, we advise against INPRR for large d's since it takes time proportional to 2^d to perturb all cells of each user. Similarly, the use of MARGRR is also hard to justify from an execution time standpoint when k gets larger, since it materializes the full marginal and applies randomized response to each cell.

Across all experiments, we find that INPHT achieves the best accuracy most consistently, and is very fast in practice.

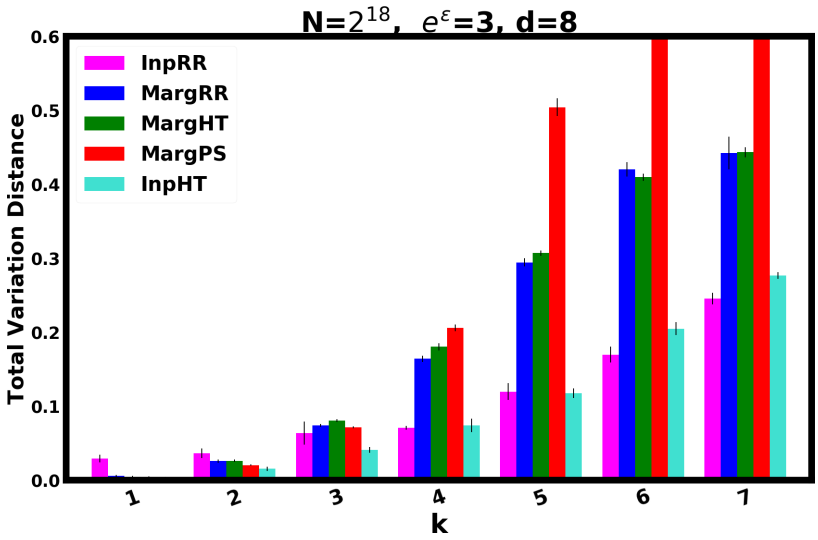


Figure 4.4: Effect of varying k.

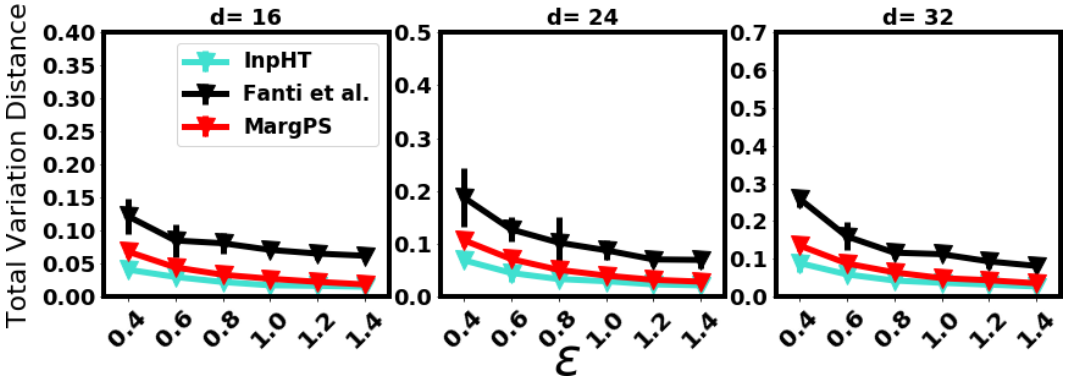


Figure 4.5: Total variation distance for $k = 2$ on NYC Taxi Trips data with $N = 2^{19}$ for larger d's.

4.4.3 Impact Of Increasing Marginal Size k

In this dissertation, our main focus has been on relatively low order marginals ($k \leq 3$), as we find this setting most compelling. However, our algorithms work for any $k \leq d$. In this section, we allow k to vary, and again measure accuracy on the taxi data set.

Experimental Setting. In this experiment, we set $N = 2^{18}, \exp(\epsilon) = 3, d = 8$ and vary k from 1 to 7 (Figure 4.4). Note that we expect to see the strongest results for InpHT when $k \leq \frac{d}{2}$; as k approaches d, we require more Hadamard coefficients, and the theoretical bound converges to that of the other input based methods.

Experimental Observations. We observe that, in line with expectations, InpHT is the method of choice for $k \leq d/2$. For larger k, InpRR appears competitive in terms of accuracy. However, there are some notable disadvantages to InpRR, as it carries with it a

much higher communication cost: the method has to send the whole input distribution, rather than a single Hadmard index and value. The aggregator's work is consequently higher as well. This ratio is 28 when $d = 8$, rising to nearly 4000 for $d = 16$. Other methods become less accurate more quickly. The absolute error does start to grow as k increases, even in the best case. However, note that a total variation distance of 0.125 in a marginal with $k = 5$ corresponds to an average absolute error of $0.125/32 \approx 0.004$ per entry.

4.4.4 Impact Of Increasing Dimensionality d

Experimental Setting. Now that we have established the relative performance of our algorithms, we compare our best methods to Fanti *et al.*(Section 4.3.4) — the only prior method available from the literature. We denote their method by INPEM. We consider a larger range of values of the dimensionality d, (achieved by duplicating columns) and show the results in Figure 4.5. For INPEM, we fix the convergence threshold to $\Omega = 0.00001$, i.e. stop when the change in the current guess is below Ω.

Experimental observations. We see that the INPEM gives reasonable results that improve as ϵ is increased. However, the achieved accuracy is several times worse than the unbiased estimators INPHT and MARGPS. There are additional reasons to not prefer INPEM: it lacks any accuracy guarantee, and so is hard to predict results. It is also slow to apply, taking several thousand or tens of thousands of iterations to converge. In some cases, the convergence criteria are immediately met by the uniform distribution, which is far from the true marginal. Weakening the convergence criterion (i.e. increasing the stopping parameter Ω) even slightly led to much worse accuracy results than the alternative methods. In contrast, our unbiased estimators are found instantaneously.

Remark. It is reasonable to ask whether EM decoding schemes can be developed for other methods for recovering marginals. The answer is allallaffirmative. The Bayes theorem in the maximization step uses the conditional probability expression of generating the noisy output γ' in response to the input γ. This probability is mechanism dependent. Therefore, we can perturb marginals using our proposals and recover them using an EM scheme. We performed a set of experiments on this approach. Our conclusion is that the EM decoder does not provide any noticeable improvement in the accuracy compared to the direct construction of unbiased estimators using the corrections proposed.

4.4.5 Impact Of Privacy Parameter ϵ

Experimental Setting. We fix N to $2^{18} \approx 0.25M$ movielens users (sampled with replacement) and change ϵ. We increase d (resp., k) along columns (rows) and vary $0.4 \leq \epsilon \leq 1.4$ to see the effect on utility in Figure 4.6.

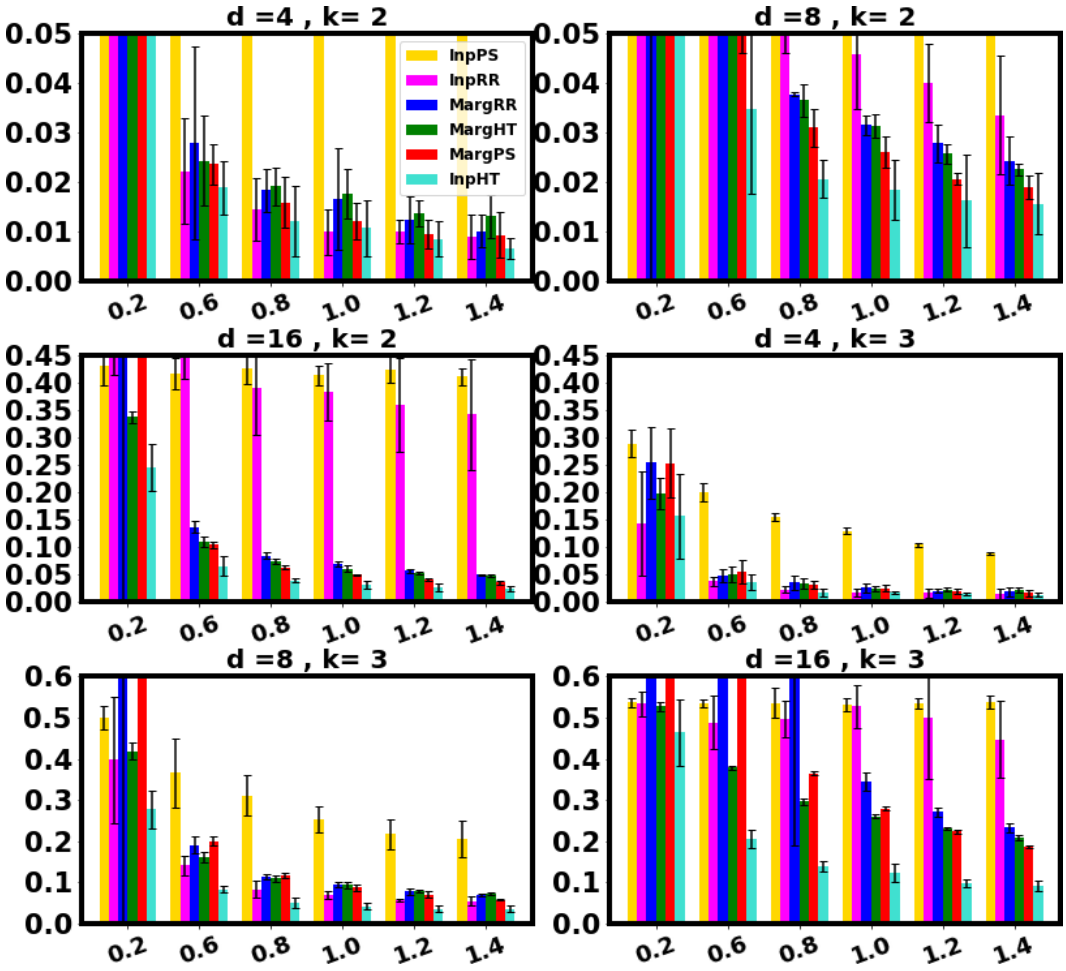


Figure 4.6: Mean total variation for $1, 2, 3-$way marginals for $N = 256K$ movielens users as ϵ varies.

Observations. We observe a decline in error as we increase the privacy budget ϵ. Once again we see that INPPS, INPRR, MARGRR are unfavourable for $k \geq 2$. MARGPS's accuracy gets better than MARGHT with increase in ϵ, although MARGHT is preferable to MARGPS for small ϵ values when d and k are larger. Yet again, INPHT consistently outperforms all other algorithms across all configurations. The main takeaway from these experiments is the confirmation that the algorithms with the best theoretical bounds on performance are borne out to be the best in practice. In general, INPHT is our first preference followed by MARGPS and MARGHT.

4.4.6 Comparison With Frequency Oracles Developed Recently

As discussed in Section 3.1.1, there have been several recent works addressing the problem of estimating population frequencies under LDP [57, 67, 72] by providing frequency oracles

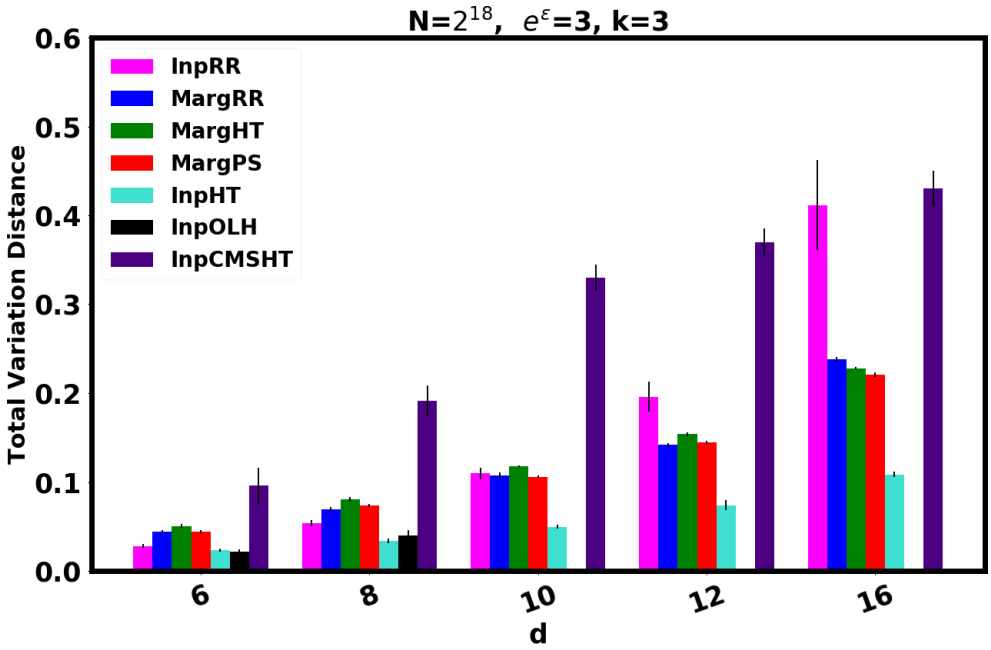


Figure 4.7: Effect of varying d with frequency oracles.

based on ideas from sketching, hashing and signal processing literature. A generic approach to marginal materialization is to build a frequency oracle, and estimate marginal probabilities by aggregating the estimated frequencies over the 2^d items from the original domain and then marginalize out the non-required attributes. In this section, we describe and compare some representative instances of this generic approach.

A key consideration of frequency oracle design is to ensure that the message sent by each user is small, compared to a possibly massive domain size. The following two approaches achieve this by hashing the input items onto a smaller domain, and applying LDP primitives to reveal information about the hashed values.

Optimized Local Hashing (INPOLH) [57]. Let's recall the OLH primitive from Section 2.4.2. OLH handles large domain size via universal hash functions. In summary, each user $i \in [N]$ with a sparse input $z_i \in \mathcal{I}_{2^m \times 2^m}$ uniformly samples a hash function h_i from a family $\mathcal{H} : [2^m] \to [1 + \exp(\epsilon)]$ of universal hash functions and hashes the signal index j_i using h_i. User i releases h_i and a noisy index j_i' perturbed using INPPS/GRR. For each user report, the aggregator has to determine the probability that the response could have come from each input value in turn, and update their beliefs accordingly. Thus, the communication cost is reduced to $\mathcal{O}(\epsilon)$ bits, but the aggregator's time cost is $\mathcal{O}(2^d)$ per user.

Private Hadamard Count-Min Sketch (INPHTCMS) [28]. The method deployed by Apple adapts ideas from sketching, and is also similar to a related method [67]. In INPHTCMS, a sketch data structure is defined with g hash functions each drawn from a family of $3-$wise independent hash functions mapping an input $j_i \in [m]$ to a much smaller

domain w. User i with a sparse input $z_i \in \mathcal{I}_{2^m \times 2^m}$ uniformly picks one of the g hash function to apply to their input, and releases a randomly sampled Hadamard coefficient of the hashed input, using randomized response. The aggregator unbiases the user reports, and uses them to reconstruct a sketch, which can be used as a frequency oracle with standard sketch estimation methods. Note that here the Hadamard transform is used to reduce the size of the communication, at the expense of a slight increase in error, in contrast to our results which use Hadmard to reduce both error and communication cost.

Experimental Setting. We set $e^\epsilon = 3$, so INPOLH hashes onto 4 possibilities. In INPHTCMS, we use $g = 5$ hash functions each of width $w = 256$ as this minimized the error observed in practice.

Experimental Observations. We applied our methods to synthetic (lightly skewed) data, and again measured total variation distance of the reconstructed marginals as we varied the dimension d (Figure 4.7). For small d, the INPOLH scheme is promising, and obtains accuracy equivalent to INPHT. However, the decoding scheme is very slow in practice, requiring the aggregator to perform a separate enumeration of the base domain for each user's response. We timed out our methods after 12 hours of computation, and so results are absent for INPOLH for the relatively small $d = 12$ and $d = 16$. While INPHTCMS is designed to accurately recover heavy hitter items (with large frequencies), it is not tuned for low-frequency items, and so is not competitive in terms of accuracy, although it is fast. Results were better when the input distribution was more skewed (results not shown). We conclude that INPHT remains the method of choice for marginal materialization under LDP.

4.5 Applications and Extensions

Since each cell of a k-way marginal is a joint distribution of a set of k attributes and can be used to determine conditional probabilities, marginals are useful in machine learning and inference tasks. In this section, following our motivational use case, we perform (1) association testing among attributes (2) dependency trees fitting. For both tasks, 1 and 2-way marginals are sufficient. Based on the accuracy results, we use MARGPS and INPHT for these tasks. Finally, we discuss how to apply our results to non-binary attributes.

4.5.1 Association Testing

Experimental setting. We use the taxi data for supporting this task since this dataset has a good mix of correlated/weakly correlated attributes (Figure 4.2). There are strong positive associations in the taxi data among the pairs ⟨Night_pick, Night_drop⟩, ⟨Toll, Far⟩ and ⟨CC, Tip⟩ and expect the test to declare them as dependent. Similarly, we expect the test to declare the pairs ⟨M_drop, CC⟩, ⟨Far, Night _pick⟩ and ⟨Toll, Night_pick⟩ to be independent.

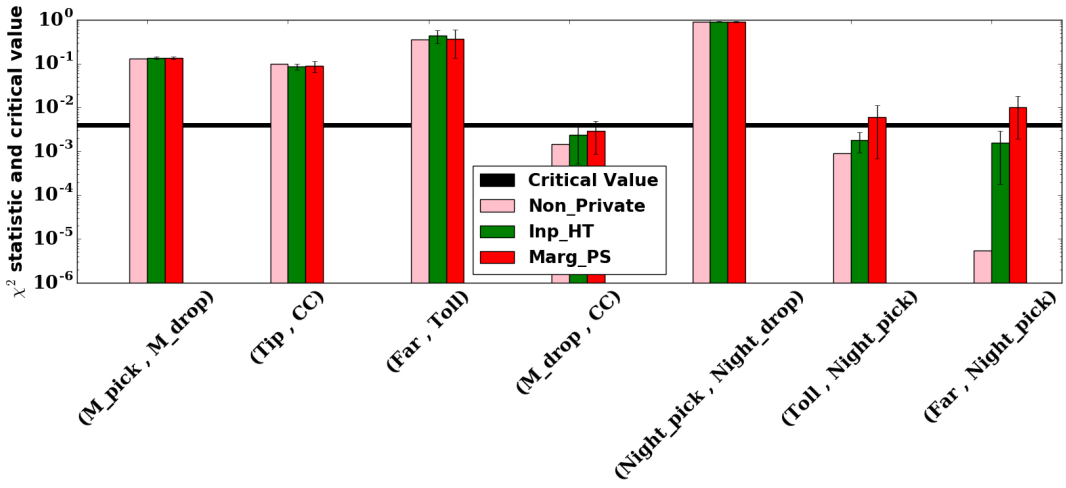


Figure 4.8: χ^2 test values on $N = 256K$ NYC taxi trips, $\epsilon = 1.1$.

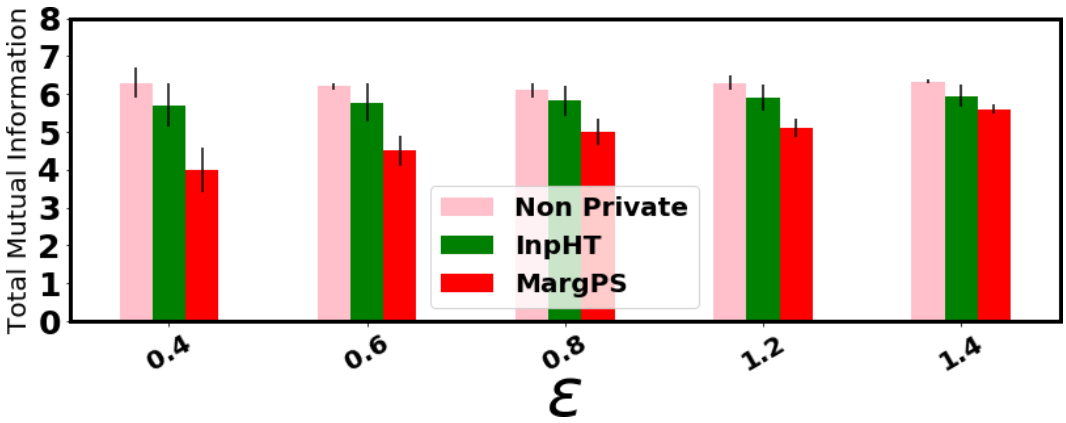


Figure 4.9: Total mutual information of trees for movielens dataset.

Experimental observations. Figure 4.8 compares privately and non-privately computed χ^2 values with the critical value (computed with 1 degree of freedom and with confidence interval of 95%[3]) over log scale. We observe that non-private and private χ^2 values are quite close in most cases for INPHT (note the log scale on the y-axis, which tends to exaggerate errors in small quantities). On the other hand, MARGPS often commits the type I error (thus failing to reject the null hypothesis) for the pairs ⟨Toll, Night_pick⟩, ⟨Far, Night_pick⟩ and occasionally for pairs ⟨M_drop , CC⟩, since the test statistic is close to the critical value in these cases.

[3]Gaboardi *et al.* in [130] suggest increasing p since comparing a differentially private χ^2 statistic to a noise unaware critical value may not lead to a good significance level even for large N. We do not perform correction in this test, and leave developing robust correlation tests under LDP for future work.

4.5.2 Bayesian Modeling

Experimental setting. Note that the Chow-Liu algorithm described in Section 2.5.1 finds a tree from the equivalence class of trees fitting the given data. Therefore, the optimal tree returned is not necessarily unique. Moreover, there could be many others trees with different topologies achieving near optimal MI score. Therefore, our aim in this section is to compare total MI from privately and non-privately learnt trees. For this purpose, we use the movielens dataset with $d = 10$.

Experimental observations. Figure 4.9 compares the total (true) MI from $200K$ users for various ϵ values (error bars show variation over different subsets of sampled records). We once again see that MI of trees computed with INPHT marginals is nearly the same as the non-private computation. MARGPS is less accurate at low ϵ's but catches up with INPHT as ϵ increases. We conclude that INPHT gives a robust solution for this approach.

4.5.3 Categoric Attributes

We now consider how to apply these methods over more general classes of input – in particular, over cases where the input is non-binary, but ranges over a larger set of possible categories $r > 2$. Suppose now we have d categoric attributes with cardinalities (indexed in order of size for convenience) $r_1 \geq r_2 \geq \ldots \geq r_d$, and wish to find marginals involving subsets of at most k attributes. We describe an approach to handling such data.

Binary encoding methods using our algorithms. Many of our algorithms such MARGRR, MARGPS, INPPS/GRR, INPRR/OUE will generalize easily in this case, since they can be applied to users represented as sparse binary vectors. The Hadamard-based methods MARGHT and INPHT can also be generalized if we rewrite the input in a binary format, i.e. we create a fresh binary attribute for each possible categoric value in an attribute (aka "one-hot encoding"). However, we can more compactly encode an attribute value that takes on r possible values using $\lceil \log_2 r \rceil$ bits, and consider this as the conjunction of $\lceil \log_2 r \rceil$ binary attributes. Consequently, we state a result (based on our strongest algorithm for the binary case) in terms of the effective binary dimension of the encoded data, $d_2 = \sum_{i=1}^{d} \lceil \log_2 r_i \rceil$; and the binary dimension of k-way marginals $k_2 = \sum_{i=1}^{k} \lceil \log_2 r_i \rceil$:

Corollary 1. *Using* INPHT *on binary encoded data, we achieve ϵ-LDP, and with constant probability we have for any target k-way marginal β on binary encoded data,*

$$\|C^\beta(t) - C^\beta(z^*)\|_1 = \tilde{O}\left(\frac{(2d_2)^{k_2/2}}{\epsilon\sqrt{N}}\right)$$

Consequently, this provides an effective solution, particularly for data with low cardinality attributes. We can see the impact of this encoding from our experiments on varying k (Figure 4.4). Observe that total variation distance over data encoded into k_2 binary

attributes is equivalent to total variation distance on binary data for a marginal of size $k = k_2$ attributes. For example, the error on a 2-way marginal over attributes with four possible values would look like that for a $k = 4$ attribute binary marginal (as in Figure 4.4).

4.6 Follow-up Works

We found two works that are directly influenced by our work [131] and solve the marginal release problem using alternative approaches.

4.6.1 Consistent Adaptive Local Marginal (CALM) [132]

Motivated from our work, the problem of aggregating marginals under LDP was revisited later by Zhang *et al.* [132]. They propose a framework consisting of a series of pre/post-processing steps to improve the overall accuracy of aggregated marginals. They construct any specified k-way marginal by combining information from multiple l-way ($l \geq k$) marginals that share the involved attributes. Specifically, they choose m sets of attributes such that any combination of k attributes are included in at least one set. These sets are chosen by modeling this task as an instance of packing and covering problem. This trick was first proposed in the centralized setting in PriView [133]. Once the sets are selected, aggregator requests users to provide noisy marginals from these sets. The marginals are then passed through a post-processing step to ensure that they sum to the same value. Finally, the target marginals are assembled based on *maximum entropy estimation* principle.

Discussion. While their solution experimentally outperforms INPHT, it is *interactive* whereas our protocol works in a single round. Besides, they do not theoretically measure the impact for their post-processing schemes on accuracy. Since the techniques from PriView do not rely on the internals of a marginal computation method, it may be possible to boost the accuracy of INPHT using these techniques.

4.6.2 ERM in Non-Interactive LDP: Efficiency and High Dimensional Case [134].

Thaler *et al.* [123] in the centralized model proposed representing histogram as a multivariate Chebyshev polynomial [135] whose coefficients can be perturbed. This approach was extended by Wang *et al.* [134] as a part of their solution to answer marginal queries. However, they do not provide any experiments and their theoretical results rely on multiple smoothness assumptions. Hence, comparing with our approach with theirs is far from immediate.

4.7 Hadamard Transformation + RR (HRR) as FO

While we targeted aggregating only the low order k-way marginals so far in chapter 4, the Hadamard transform based solution can also serve as a frequency oracle (FO) to aggregate an entire histogram when we use $k = \log_2(D) = d$. As mentioned previously, this method achieves a good compromise between accuracy and communication since each user transmits only $d + 1$ bits to describe the perturbed coefficient and its index. Though we sample and perturb only a single coefficient, this method can be generalized to any $t \leq \log_2(D)$ coefficients. Perturbing more coefficients simultaneously reduces the sampling error but increases the error due to noise and vice versa. What value of t minimizes the total error? Let's explore this question.

Generalized Hadamard Transform (HRRt). In OLH, we hash an item from a large domain $D = 2^d$ to a much smaller domain and use GRR/INPPS to perturb the hashed value. Sampling t Hadamard coefficients can also be viewed as defining a hash function $h : \{-1, 1\}^{2^d} \implies \{-1, 1\}^t$. With this anology, we can use GRR/INPPS to perturb the sampled Hadamard coefficients. We refer to this approach as HRRt.

Perturbation. User i takes the HT of z_i and samples t coefficient indices $\{j_1, j_2, .., j_t\}$ u.a.r. without replacement and perturbs $c_i = \{\phi[z_i][j_1], \phi[z_i][j_2], .., \phi[z_i][j_t]\} \in \{-1, 1\}^t$, the combination represented by t binary coefficients using GRR. User i then sends $\{j_1, j_2, .., j_t\}$ and $c'_i = \text{GRR}(c_i)$ to the aggregator.

Aggregation. The aggregator upon receiving c'_i and the index list from each user, decodes each c'_i into binary coefficients and aggregates the Hadamard coefficient array as usual. Since each $c'_i \in [2^t - 1]$ has been decoded into t binary coefficients before aggregating, we use the usual INPHT's correction with a simple change while correcting the Hadamard coefficients and not GRR's correction.

It turns out that the probability p_r of returning the truth remains the same for each sampled coefficient. For any $t \leq \log_2(D)$, we have

$$p_r = p + \frac{2^{t-1} - 1}{2^t - 1}(1 - p)$$

In the above expression, the first term $p = \frac{\exp(\epsilon)}{\exp(\epsilon)+2^t-1}$ is the probability that a combination of sampled coefficients is reported truthfully by GRR; the second term is the probability that the coefficient remains unaltered even after choosing a random combination of coefficients.

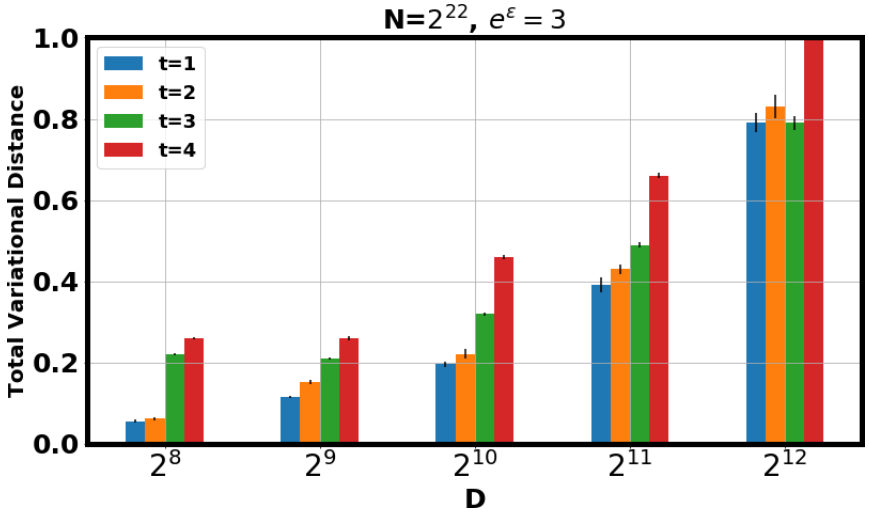


Figure 4.10: Total variation distance ($\frac{L_1}{2}$ score) as a function of t for HRRt. HRR1 offers the least variance.

Therefore,

$$\begin{aligned} p_r &= \frac{\exp(\epsilon)}{\exp(\epsilon)+2^t-1} + \frac{2^{t-1}-1}{2^t-1}\Big(1 - \frac{\exp(\epsilon)}{\exp(\epsilon)+2^t-1}\Big) \\ &= \frac{\exp(\epsilon)}{\exp(\epsilon)+2^t-1} + \frac{2^{t-1}-1}{2^t-1}\Big(\frac{2^t-1}{\exp(\epsilon)+2^t-1}\Big) \\ &= \frac{\exp(\epsilon)+2^{t-1}-1}{\exp(\epsilon)+2^t-1} \end{aligned}$$

Inheriting the notations of InpHT, any coefficient at location j can be recovered as

$$\Theta^*[j] = N_j^{-1}2^{-d}\frac{\sum_{i=1}^N H_i^*[j]}{2(\frac{\exp(\epsilon)+2^{t-1}-1}{\exp(\epsilon)+2^t-1})-1} = \frac{\sum_{i=1}^N H_i^*[j]}{2^d N_j\Big(\frac{\exp(\epsilon)-1}{\exp(\epsilon)+2^t-1}\Big)}$$

Plugging $t=1$ yields the expression from Algorithm 2. Performing calculations similar to step 2.3 in Lemma 2, for each coefficient at index j, the normalized variance is,

$$\mathrm{Var}[\mathbb{E}[\Theta^*[j]]] \leq \mathcal{O}\Big(\frac{p_r(1-p_r)}{(2p_r-1)^2}\Big) = \mathcal{O}\Big(\frac{2^{t-1}(\exp(\epsilon)+2^{t-1}-1)}{(\exp(\epsilon)-1)^2}\Big)$$

This expression shows that the variance incurred while recovering each coefficient increases with t and minimizes when $t=1$. We experimentally verify this insight. Figure 4.10 measures the error in reconstructing a synthetically generated histogram via total variation distance between the true and recovered histogram for various values of $t \in [4]$ and $D \in \{2^8, 2^9, 2^{10}, 2^{12}\}$ while fixing the $N = 2^{24}$ and $\exp(\epsilon) = 3.0$ ($\epsilon = 1.1$). Each bar is

produced after averaging 5 independent readings. We are interested in comparing bars for $t = 1, 2$. We can confirm that HRR1 outperforms HRR2 in all cases. For $D = 2^{12}$, HRR3 appears to be more accurate than HRR1 and HRR2. This behaviour can be attributed to sparsity of the original histogram. When the counts are too low to be accurately recovered, the noise levels dominate the signal and such anomalous results are observed.

Chapter 5

Range Queries

5.1 Chapter Outline And Our Contributions

In the chapter (based on [39]), we study the problem of range queries under the model of local differential privacy. Our contributions in this chapters are as follows: Our core conceptual contribution (Section 5.3) comes from proposing and analyzing several different approaches to answering one-dimensional range queries.

• We first formalize the problem and show that the flat methods — simple approach of summing a sequence of point queries entails error (measured as variance) that grows linearly with the length of the range (Section 5.3.2).

• In Section 5.4, we consider hierarchical histogram (HH) approaches, generalizing the idea of a binary tree. We show that the variance grows only logarithmically with the length of the range. Post-processing of the noisy observations can remove inconsistencies, and reduces the constants in the variance, allowing an optimal branching factor for the tree to be determined. In Section 5.4.2, we propose a post-processing scheme to further improve the accuracy the hierarchical method.

• The last approach is based on the Discrete Haar wavelet transform (described in Section 5.4.3). Here the variance is bounded in terms of the logarithm of the domain size, and no post-processing is needed. The variance bound is similar but not directly comparable to that in the hierarchical approach.

• Once we have a general method to answer range queries, we can apply it to the special case of prefix queries, and to find order statistics (medians and quantiles). We perform an empirical comparison of our methods in Section 5.5.

5.2 Model And Preliminaries

We use our usual LDP setting described in Section 2.4 and Section 4.2. For convenience, we denote the domain size by D instead of 2^d as in chapter 4. We use three representative mechanisms HRR[1] from Section 4.7, INPRR/OUE and OLH from Section 2.4.2 to implement a frequency oracle. Each one provides ϵ-LDP, by considering the probability of seeing the same output from the user if her input were to change. There are other frequency oracles mechanisms developed offering similar or weaker variance bounds (e.g. [9, 29]) and resource trade-offs but we do not include them since frequency oracles are not our main focus. Since variance for all items in the input domain for any frequency oracle F is of the same order, we denote it as V_F ($V_F = \frac{4\exp(\epsilon)}{N(\exp(\epsilon)-1)^2}$).

5.3 Range Queries

5.3.1 Problem Definition

Let's recall the definition of a range query from Section 2.2.1. We assume N non-colluding individuals each with a private item $z_i \in [D]$. For any $a < b, a \in [D], b \in [D]$, a range query $R_{[a,b]} \geq 0$ is to compute

$$R_{[a,b]} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}_{a \leq z_i \leq b}$$

where $\mathbb{I}_p$ is a binary variable that takes the value 1 if the predicate p is true and 0 otherwise.

Definition 18. *(Range Query Release Problem) Given a set of N users, the goal is to collect information guaranteeing ϵ-LDP to allow approximation of any closed interval of length $r \in [1, D]$. Let $\widehat{R}$ be an estimation of interval R of length r computed using a mechanism F. Then the quality of F is measured by the squared error $(\widehat{R} - R)^2$.*

5.3.2 Flat Solutions

One can observe that for an interval $[a, b]$, $R_{[a,b]} = \sum_{i=a}^{b} f_i$, where f_i is the (fractional) frequency of the item $i \in [D]$. Therefore a first approach is to simply sum up estimated frequencies for every item in the range, where estimates are provided by an ϵ-LDP frequency oracle: $\widehat{R}_{[a,b]} = \sum_{i=a}^{b} \widehat{\theta}_i$. We denote this approach (instantiated by a choice of frequency oracle F) as *flat* algorithms.

Fact 1. *For any range query R of length r answered using a flat method with frequency oracle F,* $\mathrm{Var}[\widehat{R} - R] = rV_F$

[1]For brevity, we denote HRR1 as HRR

Note that the variance grows linearly with the interval size which can be as large as DV_F.

Lemma 8. *The average worst case squared error over evaluation of $\binom{D}{2}$ queries $\mathcal{E}$ is $\mathcal{O}(DV_F)$.*

Proof. There are $D - r + 1$ queries of length r. Hence the average error is

$$\begin{aligned}\mathcal{E} &= \sum_{r=1}^{D} r(D-r+1)V_F \Big/ \binom{D}{2} = \frac{\left(\frac{D^2(D+1)}{2} - \frac{D(D+1)(2D+1)}{6} + D\right)V_F}{\frac{D(D-1)}{2}} \\ &= \frac{\frac{D(D+1)}{2}(D - \frac{2D+1}{3}) + D}{\frac{D(D-1)}{2}} = \frac{\left(\frac{D(D^2-1)}{6} + D\right)V_F}{\frac{D(D-1)}{2}} \\ &= \frac{D^2+5}{3(D-1)}V_F \approx \mathcal{O}(DV_F) \text{ when } D \to \infty\end{aligned}$$

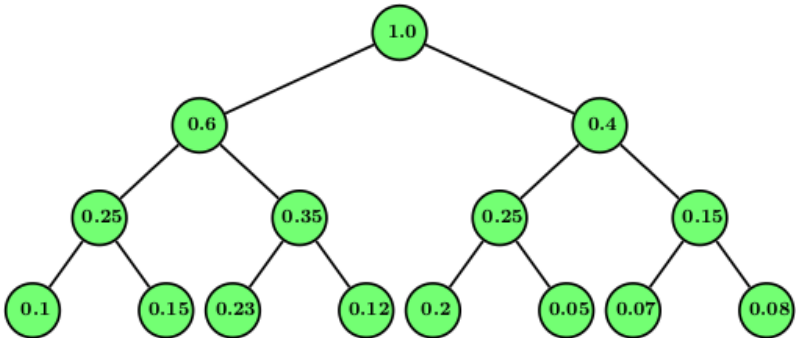
□

5.4 Hierarchical Solutions

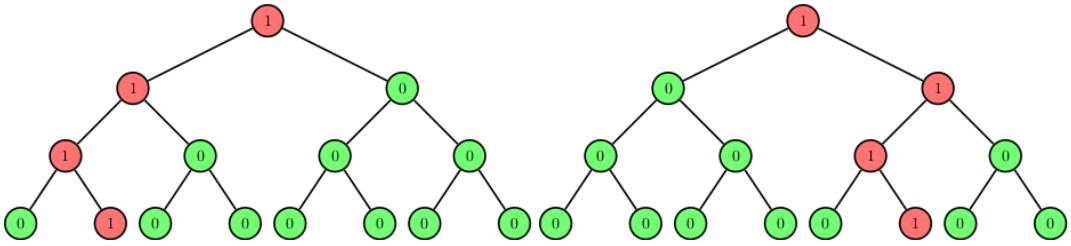
We can view the problem of answering range queries in terms of representing the frequency distribution via some collection of histograms, and producing the estimate by combining information from bins in the histograms. The "flat" approach instantiates this, and keeps one bin for each individual element. This is necessary in order to answer range queries of length 1 (i.e. point queries). However, as observed above, if we have access only to point queries, then the error grows in proportion to the length of the range. It is therefore natural to keep additional bins over subranges of the data. A classical approach is to impose a hierarchy on the domain items in such a way that the frequency of each item contributes to multiple bins of varying granularity. With such structure in place, we can answer a given query by adding counts from a relatively small number of bins. There are many hierarchical methods possible to compute histograms. Several of these have been tried in the context of centralized DP [118–121]. To the best of our knowledge, the methods that work best in centralized DP tend to rely on a complete view on the distribution, or would require multiple interactions between users and aggregator when translated to the local model. This motivates us to choose more simple yet effective strategies for histogram construction in the LDP setting. We start with the standard notion of B-adic intervals and a useful property of B-adic decompositions.

Fact 2. *For $j \in [\log_B D]$ and $B \in \mathbb{N}^+$, an interval is B-adic if it is of the form $kB^j ... (k+1)B^j - 1$ i.e. its length is a power of B and starts with an integer multiple of its length.*

Fact 3. *Any sub-range $[a, b]$ of length r from $[D]$ can be decomposed into $\leq (B-1)(2\lceil \log_B r \rceil - 1)$ disjoint B-adic ranges.*



(a) Dyadic domain decomposition with internal node weights.



(b) Local views for two users i and j ($z_i = 1$ and $z_j = 5$) with corresponding root to leaf paths marked.

Figure 5.1: An example for dyadic decomposition ($B = 2$).

For example, for $D = 32, B = 2$, the interval $[2, 22]$ can be decomposed into sub-intervals $[2, 3] \cup [4, 7] \cup [8, 15] \cup [16, 19] \cup [20, 21] \cup [22, 22]$.

The B-adic decomposition can be understood as organizing the domain under a complete B-ary tree where each node corresponds to a bin of a unique B-adic range. The root holds the entire range and the leaves hold the counts for unit sized intervals. A range query can be answered by a walk over the tree similar to the standard *pre-order traversal* and therefore a range query can be answered with at most $2(B-1)(2\lceil \log_B r \rceil - 1)$ nodes, which is at most $2(B-1)(\log_B D - 1)$ in the worst case.

5.4.1 Hierarchical Histograms (HH)

Now we describe our framework for computing hierarchical histograms. All algorithms follow a similar structure but differ on the perturbation primitive F they use:

Input transformation. User i locally arranges the input $z_i \in [D]$ in the form of a full B-ary tree of height h. Then z_i defines a unique path from a leaf to the root with a weight of 1 attached to each node on the path, and zero elsewhere. Figure 5.1 shows an example. Figure 5.1(a) shows the dyadic ($B = 2$) decomposition of the input vector $[0.1, 0.15, 0.23, 0.12, 0.2, 0.05, 0.07, 0.08]$, where the weights on internal nodes are the sum of the weights in their subtree. Figure 5.1(b) illustrates two user's local views ($z_i = 1$ and $z_j = 5$). In each local histogram, the nodes in the path from leaf to the root are shaded in

red and have a weight of 1 on each node.

Perturbation. i *samples* a level $l \in [h]$ with probability p_l. There are 2^l nodes at this level, with exactly one node of weight one and the rest zero. Hence, we can apply one of the mechanisms mentioned in Section 5.2. User i perturbs this vector using some frequency oracle F and sends the perturbed information to the aggregator along with the level id l.

Aggregation. The aggregator builds an empty tree with the same dimensions and adds the (unbiased) contribution from each user to the corresponding nodes, to estimate the fraction of the input at each node. Range queries are answered by aggregating the nodes from the B-adic decomposition of the range.

Key difference from the centralized case. Hierarchical histograms have been proposed and evaluated in the centralized case. However, the key difference here comes from how we generate information about each level. In the centralized case, the norm is to split the "error budget" ϵ into h pieces, and report the *count* of users in each node; in contrast, we have each user sample a single level, and the aggregator estimates the *fraction* of users in each node. The reason for sampling instead of splitting emerges from the analysis in Theorem 6: splitting would lead to an error proportional to h^2, whereas sampling gives an error which is at most proportional to h. Because sampling introduces some variation into the number of users reporting at each level, we work in terms of fractions rather than counts; this is important for the subsequent post-processing step.

In summary, the approach of hierarchical decomposition extends to LDP by observing the fact that it is a *linear* transformation of the original input domain. This means that adding information from the hierarchical decomposition of each individual's input yields the decomposition of the entire population. Next we evaluate the error in estimation using the hierarchical methods.

Error behaviour for Hierarchical Histograms. We begin by showing that the overall variance can be expressed in terms of the variance of the frequency oracle used, V_F. In what follows, we denote hierarchical histograms aggregated with fan-out B as HH_B.

Theorem 6. *When answering a range query of length r using a primitive F, the worst case variance V_r under the HH_B framework is $V_r \leq V_F \sum_{l=1}^{\alpha} 2(B-1)\frac{1}{p_l}$ where $\alpha = (\lceil \log_B r \rceil)$.*

Proof. Recall that all the methods we consider have the same (asymptotic) variance bound $V_F = O\big(\frac{\exp(\epsilon)}{N(\exp(\epsilon)-1)^2}\big)$, with N denoting the number of users contributing to the mechanism. Importantly, this does not depend on the domain size D, and so we can write $V_F \leq \psi_F(\epsilon)/N$, where $\psi_F(\epsilon)$ is a constant for method F that depends on ϵ. This means that once we fix the method F, the variance V_l for any node at level l will be the same, and is determined by N_l, the number of users reporting on level l. The range query $R_{[a,b]}$ of length r is decomposed

into at $2(B-1)$ nodes at each level, for $\alpha = \lceil \log_B r \rceil$ levels (from leaves upwards). So we can bound the total variance $\mathcal{V}^r$ in our estimate by

$$\sum_{l=1}^{\alpha} 2(B-1)V_l = \sum_{l=1}^{\alpha} 2(B-1)V_F/p_l = 2(B-1)V_F \sum_{l=1}^{\alpha} \frac{1}{p_l}$$

using the fact that (in expectation) $N_l = p_l N$. □

In the worst case, $\alpha = h$, and we can minimize this bound by a uniform level sampling procedure:

Lemma 9. *The quantity $\sum_{l=1}^{h} \frac{1}{p_l}$ subject to $0 \leq p_l \leq 1$ and $\sum_{l=1}^{h} p_l = 1$ is minimized by setting $p_l = \frac{1}{h}$.*

Proof. We use the Lagrange multiplier technique, and define a new function $\mathcal{L}$, introducing a new variable λ.

$$\mathcal{L}(p_1, .., p_h, \lambda) = (\sum_{l=1}^{h} \frac{1}{p_l}) + \lambda(\sum_{l=1}^{h} p_l - 1)$$

Performing partial differentiation and setting to zero, we obtain $\lambda = \frac{1}{p_1^2} = \frac{1}{p_2^2} = ... = \frac{1}{p_h^2}$ and $\sum_{l=1}^{h} \frac{1}{p_l} = 1$. Hence, $p_l = 1/\sqrt{\lambda} = 1/h$. □

Then, setting $p_l = \frac{1}{h}$ in Theorem 6 gives

$$V_r \leq 2(B-1)V_F h \lceil \log_B r \rceil. \tag{5.1}$$

Hierarchical versus flat methods. The benefit of the HH approach over the baseline flat method depends on the factor $2(B-1)h\alpha$ versus the quantity r. Note that (ignoring rounding) $h = \log_B D$ and $\alpha = \log_B r$, so we obtain an improvement over flat methods when $r > 2B \log_B^2 D$, for example. When D is very small, this may not be achieved: for $D = 64$ and $B = 2$, this condition yields $r > 128 > D$. But for larger D, e.g. $D = 2^{16}$ and $B = 2$, we obtain $r > 1024$, which equates to $\sim 1.5\%$ of the range.

Theorem 7. *The worst case average (squared) error incurred while answering all $\binom{D}{2}$ range queries using HH_B, $\mathcal{E}_B$, is (approximately) $2(B-1)V_F \log_B D \log_B \left(\frac{3D^2}{1+2D}\right)$*

Proof. We obtain the bound by summing over all range lengths r. For a given length r, there

are $D - r + 1$ possible ranges. Hence,

$$\begin{aligned}\mathcal{E}_B &\leq \frac{\sum_{r=1}^{D} V_r(D-r+1)}{D(D-1)/2} \\ &= \frac{(2(B-1)V_F \log_B D)\sum_{r=1}^{D} \log_B r(D-r+1)}{D(D-1)/2} \\ &= \frac{2(B-1)V_F \log_B D\Big[(D+1)\log_B(\prod_{r=1}^{D} r) - \sum_{r=1}^{D} \log_B r^r\Big]}{D(D-1)/2}\end{aligned}$$

We find bounds on each of the two components separately.

1. Using Stirling's approximation we have $\log_B D! \leq \log_B(D^{(D+\frac{1}{2})}\exp(1-D)) < (D+1)\log_B D$.

2. Writing $P = \sum_{r=1}^{D} r = D(D+1)/2$ and $Q = \sum_{r=1}^{D} r^2 = D(D+1)(2D+1)/6$, we make use of Jensen's inequality to get

$$\begin{aligned}\sum_{r=1}^{D} r\log_B r = P\sum_{r=1}^{D}\frac{r}{P}\log_B r &\leq P\log_B(\sum_{r=1}^{D} r\frac{r}{P}) \\ &= P\log_B(Q/P) = D(D+1)/2\log_B\Big(1+2D/3\Big)\end{aligned}$$

Plugging these upper bounds in to the main expression,

$$\begin{aligned}\mathcal{E}_B &< \frac{2(B-1)V_F\log_B D\Big[(D+1)^2\log_B D - \frac{D(D+1)}{2}\log_B\Big(\frac{1+2D}{3}\Big)\Big]}{D(D-1)/2} \\ &= 2(B-1)V_F\log_B D\Big[\frac{2(D+1)^2\log_B D}{D(D-1)} - \frac{D+1}{D-1}\log_B\Big(\frac{1+2D}{3}\Big)\Big] \\ &\approx 2(B-1)V_F\log_B D\log_B\Big(\frac{3D^2}{1+2D}\Big)\text{as } D\to\infty.\end{aligned}$$

□

Key difference from the centralized case. Similar looking bounds are known in centralized case, for example due to Qardaji *et al.* [119], but with some key differences. There, the bound (simplified) is proportional to $(B-1)h^3V_F$ rather than the $(B-1)h^2V_F$ we see here.

The difference arises because [119] scales the parameter ϵ by a factor of h, which introduces the factor of $h \cdot h^2 = h^3$ into the variance; in contrast, sampling each level with probability $1/h$ scales the variance only by h^2. Note however that in the centralized case V_F scales proportionate to $1/N^2$ rather than $1/N$ in the local case: a necessary cost to provide local privacy guarantees.

Optimal branching factor for HH$_B$. In general, increasing the fan-out has two consequences under our algorithmic framework. Large B reduces the tree height, which

increases accuracy of estimation per node since larger population is allocated to each level. But this also means that we can require more nodes at each level to evaluate a query which tends to increase the total error incurred during evaluation. We would like to find a branching factor that balances these two effects. We use the expression for the variance in (5.1) to find the optimal branching factor for a given D. We first compute the gradient of the function $2(B-1)\log_B(r)\log_B(D)$. Differentiating w.r.t. B we get

$$\begin{aligned}\nabla &= \frac{\partial}{\partial B}\Big[\frac{2(B-1)\ln(D)\ln(r)}{\ln^2 B}\Big] = 2\ln D\ln r\frac{\partial}{dB}\Big[\frac{B-1}{\ln^2 B}\Big] \\ &= \frac{2\ln(D)\ln(r)}{(\ln^2 B)^2}\big(\ln^2 B\frac{\partial}{\partial B}[B-1] - (B-1)\frac{\partial}{\partial B}[\ln^2 B]\big) \\ &= 2\ln D\ln r\big(\ln^2 B - \frac{2}{B}(B-1)\ln B\big)/\ln^4 B \\ &= 2\ln D\ln r(B\ln B - 2B + 2)/B\ln^3 B\end{aligned}$$

We now seek a B such that the derivative $\nabla = 0$. The numerical solution is (approximately) $B = 4.922$. Hence we minimize the variance by choosing B to be 4 or 5. This is again in contrast to the centralized case, where the optimal branching factor is determined to be approximately 16 [119].

5.4.2 Post-processing for consistency

There is some redundancy in the information materialized by the HH approach: we obtain estimates for the weight of each internal node, as well as its child nodes, which should sum to the parent weight. We observe that the accuracy of the HH framework can be further improved by finding the *least squares* solution for the weight of each node taking into account all the information we have about it, i.e. for each node v, we approximate the (fractional) frequency $f(v)$ with $\widehat{f}(v)$ such that $||f(v) - \widehat{f}(v)||_2$ is minimized subject to the consistency constraints. We can invoke the Gauss-Markov theorem since the variance of all our estimates are equal, and hence the least squares solution is the best linear unbiased estimator.

Lemma 10. *The least-squares estimated counts reduce the associated variance by a factor of at least $\frac{B}{B+1}$ in a hierarchy of fan-out B.*

Proof. We begin by considering the linear algebraic formulation. Let H denote the $n \times D$ matrix that encodes the hierarchy, where n is the number of nodes in the tree structure. For instance, if we consider a single level tree with B leaves, then $H = \begin{bmatrix} \mathbf{1}_D \\ I_D \end{bmatrix}$, where $\mathbf{1}_D$ is the D-length vector of all 1s, and $\mathcal{I}_D$ is the $D \times D$ identity matrix. Let $\mathbf{x}$ denote the vector of reconstructed (noisy) frequencies of nodes. Then the optimal least-squares estimate of

the true counts can be written as $\widehat{\mathbf{c}} = (H^T H)^{-1} H^T \mathbf{x}$. Denote a range query $R_{[a,b]}$ as the length D vector that is 1 for indices between a and b, and 0 otherwise. Then the answer to our range query is $R_{[a,b]}^T \widehat{\mathbf{c}}$. The variance associated with query $R_{[a,b]}$ is given by

$$\begin{aligned}
\mathrm{Var}[R_{[a,b]}^T \widehat{\mathbf{c}}] &= \mathrm{Var}[R_{[a,b]}^T (H^T H)^{-1} H^T \mathbf{x}] \\
&= R_{[a,b]}^T (H^T H)^{-1} H^T \mathrm{Cov}(\mathbf{x}) H ((H^T H)^{-1})^T R_{[a,b]} \\
&= R_{[a,b]}^T (H^T H)^{-1} H^T V_F I_D H ((H^T H)^{-1})^T) R_{[a,b]} \\
&= V_F R_{[a,b]}^T (H^T H)^{-1} (H^T H) ((H^T H)^{-1})^T) R_{[a,b]} \\
&= V_F R_{[a,b]}^T (H^T H)^{-1} R_{[a,b]}
\end{aligned}$$

First, consider the simple case when H is a single level tree with B leaves. Then we have $H^T H = \mathbf{1}_{B \times B} + \mathcal{I}_B$, where $\mathbf{1}_{B \times B}$ denotes the $B \times B$ matrix of all ones. We can verify below that $(H^T H)^{-1} = \mathcal{I}_B - \frac{\mathbf{1}_{B \times B}}{B+1}$.

$$\begin{aligned}
\Big(\mathcal{I}_B - \frac{\mathbf{1}_{B \times B}}{B+1}\Big) H^T H &= \Big(\mathcal{I}_B - \frac{\mathbf{1}_{B \times B}}{B+1}\Big)\Big(\mathbf{1}_{B \times B} + \mathcal{I}_B\Big) \\
&= \mathcal{I}_B \times \mathbf{1}_{B \times B} + \mathcal{I}_B - \frac{\mathbf{1}_{B \times B} \times \mathbf{1}_{B \times B}}{B+1} - \frac{\mathbf{1}_{B \times B} \times \mathcal{I}_B}{B+1} \\
&= \mathbf{1}_{B \times B} + \mathcal{I}_B - \frac{B}{B+1} \mathbf{1}_{B \times B} - \frac{1}{B+1} \mathbf{1}_{B \times B} \\
&= \mathcal{I}_B + \Big(1 - \frac{B+1}{B+1}\Big) \mathbf{1}_{B \times B} = \mathcal{I}_B
\end{aligned}$$

From this we can quickly read off the variance of any range query. For a point query, the associated variance is simply $B/(B+1)V_F$, while for a query of length r, the variance equates to $(rB - r(r-1))/(B+1)V_F$. Observe that the variance for the whole range $r = B$ is just $B/(B+1)V_F$, and that the maximum variance is for a range of just under half the length, $r = (B+1)/2$, which gives a bound of

$$V_F(B+1)(B+1)/(4(B+1)) = (B+1)V_F/4.$$

The same approach can be used for hierarchies with more than one level. However, while there is considerable structure to be studied here, there is no simple closed form, and forming $(H^T H)^{-1}$ can be inconvenient for large D. Instead, for each level, we can apply the argument above between the noisy counts for any node and its B children. This shows that if we applied this estimation procedure to just these counts, we would obtain a bound of $B/(B+1)V_F$ to any node (parent or child), and at most $(B+1)V_F/4$ for any sum of node counts. Therefore, if we find the optimal least squares estimates, their (minimal) variance can be at most this much. □

Consequently, after this constrained inference, the error variance at each node

is at most $\frac{BV_F}{B+1}$. It is possible to give a tighter bound for nodes higher up in the hierarchy: the variance reduces by $\frac{B^i}{\sum_{j=0}^{i} B^j}$ for level i (counting up from level 1, the leaves). This approaches $\frac{B-1}{B}$, from above; however, we adopt the simpler $\frac{B}{B+1}$ bound for clarity.

This modified variance affects the worst case error, and hence our calculation of an optimal branching factor. From the above proof, we can obtain a new bound on the worst case error of $(B+1)V_F/2$ for every level touched by the query (that is, $(B+1)V_F/4$ for the left and right fringe of the query). This equates to $(B+1)V_F \log_B(r) \log_B(D)/2$ total variance. Differentiating w.r.t. B, we find

$$\begin{aligned}\nabla &= \frac{\partial}{\partial B}\Big[(B+1)\log_B(r)\log_B(D)V_F/2\Big] \\ &= \ln(r)\ln(D)(B\ln B - 2B - 2)/B\ln^3 B\end{aligned}$$

Consequently, the value that minimizes ∇ is $B \approx 9.18$ — larger than without consistency. This implies a constant factor reduction in the variance in range queries from post-processing. Specifically, if we pick $B = 8$ (a power of 2), then this bound on variance is

$$9V_F \log_2(r)\log_2(D)/(2\log_2^2 8) = \frac{1}{2}V_F \log_2(r)\log_2(D), \tag{5.2}$$

compared to $\frac{7}{4}V_F \log_2(r)\log_2(D)$ for HH_4 without consistency. We confirm this reduction in error experimentally in Section 5.5.

We can make use of the structure of the hierarchy to provide a simple linear-time procedure to compute optimal estimates. This approach was introduced in the centralized case by Hay *et al.* [118]. Their efficient two-stage process can be translated to the local model.

Stage 1: Weighted Averaging. Traversing the tree bottom up, we use the weighted average of a node's original reconstructed frequency $f(.)$ and the sum of its children's (adjusted) weights to update the node's reconstructed weight. For a non-leaf node v, its adjusted weight is a weighted combination as follows:

$$\bar{f}(v) = \frac{B^i - B^{i-1}}{B^i - 1} f(v) + \frac{B^{i-1} - 1}{B^i - 1} \sum_{u \in \text{child}(v)} \bar{f}(u)$$

Stage 2: Mean Consistency. This step makes sure that for each node, its weight is equal to the sum of its children's values. This is done by dividing the difference between parent's weight and children's total weight equally among children. For a non-root node v,

$$\widehat{f}(v) = \bar{f}(v) + \frac{1}{B}\Big[\widehat{f}(p(v)) - \sum_{u \in \text{child}(v)} \bar{f}(u)\Big]$$

$$\frac{1}{\sqrt{8}}\begin{pmatrix} 1 & 1 & \sqrt{2} & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & \sqrt{2} & 0 & -2 & 0 & 0 & 0 \\ 1 & 1 & -\sqrt{2} & 0 & 0 & 2 & 0 & 0 \\ 1 & 1 & -\sqrt{2} & 0 & 0 & -2 & 0 & 0 \\ 1 & -1 & 0 & \sqrt{2} & 0 & 0 & 2 & 0 \\ 1 & -1 & 0 & \sqrt{2} & 0 & 0 & -2 & 0 \\ 1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & 2 \\ 1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & -2 \end{pmatrix}$$

Figure 5.2: DHT matrix for $D = 8$.

where $\bar{f}(p(v))$ is the weight of v's parent after weighted averaging. The values of $\widehat{f}$ achieve the minimum L_2 solution.

Finally, we note that the cost of this post-processing is relatively low for the aggregator: each of the two steps can be computed in a linear pass over the tree structure. A useful property of finding the least squares solution is that it enforces the consistency property: the final estimate for each node is equal to the sum of its children. Thus, it does not matter how we try to answer a range query (just adding up leaves, or subtracting some counts from others) — we will obtain the same result.

Key difference from the centralized case. Our post-processing is influenced by a sequence of works in the centralized case. However, we do observe some important points of departure. First, because users sample levels, we work with the distribution of frequencies across each level, rather than counts, as the counts are not guaranteed to sum up exactly. Secondly, our analysis method allows us to give an upper bound on the variance at every level in the tree – prior work gave a mixture of upper and lower bounds on variances. This, in conjunction with our bound on covariances allows us to give a tighter bound on the variance for a range query, and to find a bound on the optimal branching factor after taking into account the post-processing, which has not been done previously.

5.4.3 Discrete Haar Transform (DHT)

The Discrete Haar Transform (DHT) provides an alternative approach to summarizing data for the purpose of answering range queries. DHT is a popular data synopsis tool that relies on a hierarchical (binary tree-based) decomposition of the data. DHT can be understood as performing recursive pairwise averaging and differencing of our data at different granularities, as opposed to the HH approach which gathers sums of values. The method imposes a full binary tree structure over the domain, where $h(v)$ is the height of node v, counting up from the leaves (level 0). The Haar coefficient c_v for a node v is computed as $c_v = \frac{C_l - C_r}{2^{h(v)/2}}$, where

C_l, C_r are the sum of counts of all leaves in the left and right subtree of v. In the local case when z_i represents a leaf of the tree, there is exactly one non-zero Haar coefficient at each level l with value $\pm\frac{1}{2^{l/2}}$. The DHT can also be represented as a matrix H_D of dimension $D \times D$ (where D is a power of 2) with each row j encoding the Haar coefficients for item $j \in [D]$. We can decode the count at any leaf node v by taking the inner product of the vector of Haar coefficients with the row of H_D corresponding to v. Observe that we only need h coefficients to answer a point query.

Answering a range query. A similar fact holds for range queries. We can answer any range query by first summing all rows of H_D that correspond to leaf nodes within the range, then taking the inner product of this with the coefficient vector. We can observe that for an internal node in the binary tree, if it is fully contained (or fully excluded) by the range, then it contributes zero to the sum. Hence, we only need coefficients corresponding to nodes that are cut by the range query: there are at most $2h$ of these. The main benefit of DHT comes from the fact that all coefficients are independent, and there is no redundant information. Therefore we obtain a certain amount of consistency by design: any set of Haar coefficients uniquely determines an input vector, and there is no need to apply the post-processing step described in Section 5.4.2.

Our algorithmic framework. For convenience, we rescale each coefficient reported by a user at a non-root node to be from $\{-1, 0, 1\}$, and apply the scaling factor later in the procedure. Similar to the HH approach, each user samples a level l with probability p_l and perturbs the coefficients from that level using a suitable perturbation primitive. Each user then reports her noisy coefficients along with the level. The aggregator, after accepting all reports, prepares a similar tree and applies the correction to make an unbiased estimation of each Haar coefficient. The aggregator can evaluate range queries using the (unbiased but still noisy) coefficients.

Perturbing Haar coefficients. As with hierarchical histogram methods, where each level is a sparse (one hot) vector, there are several choices for how to release information about the sampled level in the Haar tree. The only difference is that previously the non-zero entry in the level was always a 1 value; for Haar, it can be a -1 or a 1. There are various straightforward ways to adapt the methods that we have already (see, for example, [9, 78, 129]). We choose to adapt the Hadamard Randomized Response (HRR) method, described in Section 4.7. First, this is convenient: it immediately works for negative valued weights without any modification. But it also minimizes the communication effort for the users: they summarize their whole level with a single bit (plus the description of the level and Hadmard coefficient chosen). We have confirmed this choice empirically in calibration experiments (omitted for brevity): HRR is consistent with other choices in terms of accuracy, and so is preferred for its convenience and compactness.

Recall that the (scaled) Hadamard transform of a sparse binary vector e_i is equivalent to selecting the ith row/column from the Hadamard matrix. When we transform $-e_i$, the Hadamard coefficients remain binary, with their signs negated. Hence we use HRR for perturbing levelwise Haar coefficients. At the root level, where there is a single coefficient, this is equivalent to 1 bit RR. The 0th wavelet coefficient c_0 can be hardcoded to $\frac{N}{D}$ since it does not require perturbation. We refer to this algorithm as HaarHRR.

Error behaviour for HaarHRR. As mentioned before, we answer an arbitrary query of length r by taking a weighted combination of at most $2h$ coefficients. A coefficient u at level $l(u)$ contributes to the answer if and only if exactly one of the leftmost and rightmost leaves of the subtree of node u intersects with the range. The 0th coefficient c_0 is assigned the weight r. Let O_u^L (O_u^R) be the size of the overlap sets for left (right) subtree for u with the range. Using reconstructed coefficients, we evaluate a query to produce answer $\widehat{R}$ as:

$$\widehat{R} = rc_0 + \sum_u \Big(\frac{O_u^L - O_u^R}{2^{l(u)}}\Big)\widehat{c}_u$$

where, $\widehat{c}_u$ is an unbiased estimation of a coefficient c_u at level $l(u)$. In the worst case, the absolute weight $|O_u^L - O_u^R| = 2^{l(u)-1}$. We can analyze the corresponding varance, V_r, by observing that there at most two coefficients used in each level:

$$V_r \leq 2\sum_{l=1}^{h}\Big(\frac{2^{l-1}}{2^l}\Big)^2 V_F = \sum_{l=1}^{h}\frac{1}{2}V_F = \frac{1}{2}\sum_{l=1}^{h}\frac{V_F}{p_l}$$

Here, V_F is the variance associated with the HRR frequency oracle. As in the hierarchical case, the optimal choice is to set $p_l = 1/h$ (i.e. we sample a level uniformly), where $h = \log_2(D)$. Then we obtain

$$V_r = \frac{1}{2}\log_2^2(D)V_F \tag{5.3}$$

It is instructive to compare this expression with the bounds obtained for the hierarchical methods. Recall that, after post-processing for consistency, we found that the variance for answering range queries with HH_8, based on optimizing the branching factor, is $\log_2(r)\log_2(D)V_F/2$ (from (5.2)). That is, for long range queries where r is close to D, (5.3) will be close to (5.2). Consequently, we expect both methods to be competitive, and will use empirical comparison to investigate their behaviour in practice.

Finally, observe that since this bound does not depend on the range size itself, the average error across all possible range queries is also bounded by (5.3).

Key difference from the centralized case. The technique of perturbing Haar coefficients to answer differentially private range queries was proposed and studied in the centralized case

under the name "privelets" [117]. Subsequent work argued that more involved centralized algorithms could obtain better accuracy. We will see in the experimental section that HaarHRR is among our best performing methods. Hence, our contribution in this work is to reintroduce the DHT as a useful tool in local privacy.

5.4.4 Prefix and Quantile Queries

Prefix queries form an important class of range queries, where the start of the range is fixed to be the first point in the domain. We only consider the prefix queries with the left end fixed to the zeroth item of the domain i.e. 0. Let's recall the definition of a prefix query.

$$R_{[0,b]} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_{0 \leq z_i \leq b}$$

The methods we have developed allow prefix queries to be answered as a special case. Note that for hierarchical and DHT-based methods, we expect the error to be lower than for arbitrary range queries. Considering the error in hierarchical methods (Theorem 6), we require at most $B-1$ nodes at each level to construct a prefix query, instead of $2(B-1)$, which reduces the variance by almost half. For DHT similarly, we only split nodes on the right end of a prefix query, so we also reduce the variance bound by a factor of 2. Note that a reduction in variance by 0.5 will translate into a factor of $\sqrt{2} = 0.707$ in the absolute error. Although the variance bound changes by a constant factor, we obtain the same optimal choice for the branching factor in B.

Prefix queries are sufficient to answer quantile queries. The ϕ-quantile is the index j in the domain such that at most a ϕ-fraction of the input data lies below j, and at most a $(1-\phi)$ fraction lies above it. If we can pose arbitrary prefix queries, then we can binary search for a prefix j such that the prefix query on j meets the ϕ-quantile condition. Errors arise when the noise in answering prefix queries causes us to select a j that is either too large or too small. The quantiles then describe the input data distribution in a general purpose, non-parametric fashion. Our expectation is that our proposed methods should allow more accurate reconstructions of quantiles than flat methods, since we expect they will observe lower error. We formalize the problem:

Definition 19. *(Quantile Query Release Problem) Given a set of N users, the goal is to collect information guaranteeing ϵ-LDP to approximate any quantile $q \in [0,1]$. Let $\widehat{Q}$ be the item returned as the answer to the quantile query q using a mechanism F, which is in truth the q' quantile, and let Q be the true q quantile. We evaluate the quality of F by both the* value error*, measured by the squared error $(\widehat{Q} - Q)^2$; and the* quantile error *$|q - \widehat{q}|$.*

Dataset	Type	Bucketing attribute	Time span	Bucket size	N	D
stackoverflow [136]	time series	posting/editing answers	2774 days	12 min.	$\approx 2^{21}$	2^{18}
movielens [128]	time series	user rating	random sample of users from 1995 to 2018	32 min.	$\approx 2^{21}$	2^{18}
NYC yellow taxi dataset [126]	time series	pickup time	9/2017 to 12/2017	30 min.	$\approx 2^{24}$	2^{17}
Gowalla [137]	location details	check-in co-ordinates	02/2009 to 09/2010	geohash of length 5 ($\pm$2.4km error)	$\approx 2^{21}$	2^{16}

Table 5.1: Summary of datasets used.

5.5 Experimental Evaluation

Our goal in this section is to validate our solutions and theoretical claims with experiments. We first test on synthetic data and then use real world datasets with our best performing methods. To the best of our knowledge, since the problem of range/quantile queries has not been dealt before in the local setting, we do not have any prior results to compare with.

Synthetic Dataset. We are interested in comparing the flat, hierarchical and wavelet methods for range queries of varying lengths on large domains, capturing meaningful real-world settings. Generating independent samples from well-known distributions conveniently allow us to control the shape of input histograms. We have evaluated the methods over a variety of real and synthetic data. Our observation is that measures such as speed and accuracy do not depend too heavily on the data distribution. Hence, we present here results on synthetic data sampled from Cauchy distributions. This allows us to easily vary parameters such as the population size N and the domain size D, as well as varying the distribution to be more or less skewed. We vary the domain size D from small ($D = 2^8$) to large ($D = 2^{22}$) as powers of two.

Real Datasets. We use three popular time-series datasets and a location dataset summarized in Table 5.1. In the time-series datasets, we divide the total timespan into slots of a fixed length and bucketize the records at a suitably fine grain for queries, while ensuring that the histogram have *heavy* intervals with large amounts of mass concentrated. For location data, a standard hierarchical way of encoding GPS co-ordinates into a fixed length signature is to geohash them [138]. The hash length determines the coarseness of a bucket. Points sharing a common prefix are in a close proximity and included in a rectangle of that prefix. The shorter a geohash is, the larger its rectangle.

Algorithm default parameters and settings. We set a default value of $\exp(\epsilon) = 3$ ($\epsilon = 1.1$), in line with prior work on LDP. This means, for example, that binary randomized response will report a true answer $\frac{3}{4}$ of the time, and lie $\frac{1}{4}$ of the time — enough to offer plausible deniability to users, while allowing algorithms to achieve good accuracy. Since the domain size D is chosen to be a power of 2, we can choose a range of branching factors B for hierarchical histograms so that $\log_B(D)$ remains an integer. The default population

size N is set to be $N = 2^{26}$ which captures the scenario of an industrial deployment, similar to [12, 28, 68]. Each bar plot is the mean of 5 repetitions of an experiment and error bars capture the observed standard deviation. The simulations are implemented in C++ and tested on a standard Linux machine. To the best of our knowledge, ours is among the first non-industrial work to provide simulations with domain sizes as large as 2^{22}. Our implementation is publicly available [139].

Sampling range queries for evaluation. When the domain size is small or moderate ($D = 2^8$ and 2^{16}), it is feasible to evaluate all $\binom{D}{2}$ range queries and their exact average. However, this is not scalable for larger domains, and so we average over a subset of the range queries. To ensure good coverage of different ranges, we pick a set of evenly-spaced starting points, and then evaluate all ranges that begin at each of these points. For $D = 2^{17}, 2^{18}, 2^{20}, 2^{21}$ and 2^{22} we pick start points every $2^8, 2^{10}, 2^{14}, 2^{16}$ and 2^{17} steps, respectively, yielding a total of 33.3M and 67.1M unique queries.

Histogram estimation primitives. The HH framework in general is agnostic to the choice of the histogram estimation primitive F. We show results with OUE, HRR and OLH as the primitives for histogram reconstruction, since they are considered to be state of art, and all provide the same theoretical bound V_F on variance. Though any of these three methods can serve as a flat method, we choose OUE as a flat method since it can be simulated efficiently and reliably provides the lowest error in practice by a small margin. We refer to the hierarchical methods using HH framework as TreeOUE, TreeOLH and TreeHRR. Their counterparts where the aggregator applies post-processing to enforce consistency are identified with the CI (Constraint Inference) suffix, e.g. TreeHRRCI.

We quickly observed in our preliminary experiments that direct implementation of OUE can be very slow for large D: the method perturbs and reports D bits for each user. For accuracy evaluation purposes, we can replace the slow method with a statistically equivalent simulation. That is, we can simulate the aggregated noisy count data that the aggregator would receive from the population. We know that noisy count of any item is aggregated from two distributions (1) "true" ones that are reported as ones (with prob. $\frac{1}{2}$) (2) zeros that are flipped to be ones (with prob. $\frac{1}{1+\exp(\epsilon)}$). Therefore, using the (private) knowledge of the true count $\theta[j]$ of item $j \in [D]$, the noisy count $\theta^*[j]$ can be expressed as a sum of two binomial random variables,

$$\theta^*[j] = \text{Binomial}(\theta[j], 0.5) + \text{Binomial}\Big(N - \theta[j], \frac{1}{1+\exp(\epsilon)}\Big)$$

Our simulation can perform this sampling for all items, then provides the sampled count to the aggregator, which then performs the usual bias correction procedure.

The OLH method suffers from a more substantial drawback: the method is very slow for the aggregator to decode, due to the need to iterate through all possible inputs for

each user report (time $O(ND)$). We know of no short cuts here, and so we only consider OLH for our initial experiments with small domain size D.

5.5.1 Impact of varying B and r

Experiment description. In this experiment, we aim to study how much a privately reconstructed answer for a range query deviates from the ground truth. Each query answer is normalized to fall in the range 0 to 1, so we expect good results to be much smaller than 1. To compare with our theoretical analysis of variance, we measure the accuracy in the form of mean squared error between true and reconstructed range query answers.

Plot description. Figure 5.3 illustrates the effect of branching factor B on accuracy for domains of size 2^8 (small), 2^{16} (medium), and 2^{22} (large). Within each plot with a fixed D and query length r, we vary the branching factor on the X axis. We plot the flat OUE method as if it were a hierarchical method with $B = D$, since it effectively has this fan out from the root. We treat HaarHRR as if it has $B = 2$, since is based on a binary tree decomposition. The Y axis in each plot shows the mean squared error incurred while answering all queries of length r. As the plots go top to bottom, the range length in each column increases from 1 to a constant fraction of the whole domain size D. The leftmost column of plots have $D = 2^8$, and the rightmost column of plots have $D = 2^{22}$.

Observations. Our first observation is that the CI step reliably provides a significant improvement in accuracy in almost all cases for HH, and never increases the error. Our theory suggests that the CI step improves the worst case accuracy by a constant factor, and this is borne out in practice. This improvement is more pronounced at larger intervals and higher branching factors. In many cases, especially in the right three columns, TreeOUECI and TreeHRRCI are two to four times more accurate then their inconsistent counter parts. Consequently, we put our main focus on methods with consistency applied in what follows.

Next, we quickly see evidence that the flat approach (represented by OUE) is not effective for answering range queries. Unsurprisingly, for point queries ($r = 1$), flat methods are competitive. This is because all methods need to track information on individual item frequencies, in order to answer short range queries. The flat approach keeps only this information, and so maximizes the accuracy here. Meanwhile, HH methods only use leaf level information to answer point queries, and so we see better accuracy the shallower the tree is, i.e. the bigger B is. However, as soon as the range goes beyond a small fraction of the domain size (ranges in the few tens in length), other approaches are preferable. The second column of plots shows results for relatively short ranges where the flat method is not the most accurate. Note that our methods as proposed are agnostic as to the workload of range queries, and optimize across all range queries. If a workload were known, we could

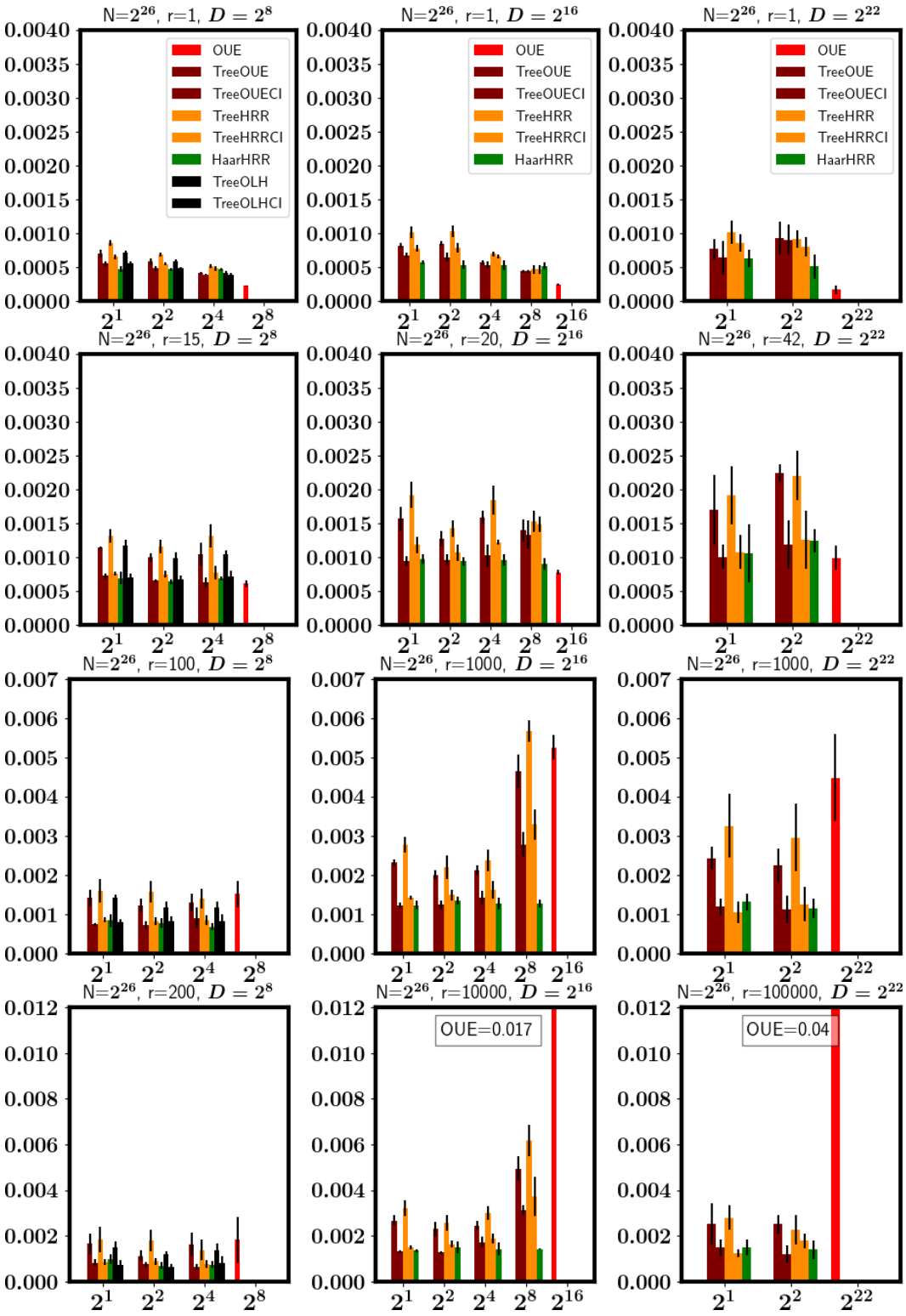


Figure 5.3: Impact of post-processing and branching factor B. In each plot, B increases along X axis, and the Y axis gives the MSE for all range queries of length r. The second row corresponds to the range size where HaarHRR outperforms the flat method.

easily optimize for this by adjusting the sampling probabilities p_l of the HH methods, for example to give more accuracy on short queries if needed.

For larger domain sizes and queries, our methods outperform the flat method by a high margin. For example, the best hierarchical methods for very long queries and large domains are at least 16 times more accurate than the flat method. Recall our discussion of OLH above that emphasized that its computation cost scales poorly with domain size D. We show results for TreeOLH and TreeOLHCI for the small domain size 2^8, but drop them for larger domain sizes, due to this poor scaling. We can observe that although the method achieves competitive accuracy, it is equalled or beaten by other more performant methods, so we are secure in omitting it.

As we consider the two tree methods, TreeOUE and TreeHRR, we observe that they have similar patterns of behaviour. In terms of the branching factor B, it is difficult to pick a single particular B to minimize the variance, due to the small relative differences. The error seems to decrease from $B = 2$, and increase for larger B values above 2^4 (i.e. 16). Across these experiments, we observe that choosing $B = 4$, 8 or 16 consistently provides the best results for medium to large sized ranges. This agrees with our theory, which led us to favour $B = 8$ or $B = 4$, with or without consistency applied respectively. This range of choices means that we are not penalized severely for failing to choose an optimal value of B.

The main takeaway from Figure 5.3 is the strong performance for the HaarHRR method. It is not competitive for point queries ($r = 1$), but for all ranges except the shortest it achieves the single best or equal best accuracy. For some of the long range queries covering the almost the entire domain, it is slightly outperformed by consistent HH_B methods. However, this is sufficiently small that it is hard to observe visually on the plots. Across a broad range of query lengths (roughly, 0.1% to 10% of the domain size), HaarHRR is preferred. It is most clearly the preferred method for smaller domain sizes, such as in the case of $D = 2^8$. We observed a similar behaviour for domains as small as 2^5.

5.5.2 Impact of privacy parameter ϵ

Experiment description. We now vary ϵ between 0.1 (higher privacy) to 1.4 (lower privacy) and find the mean squared error over range queries. Similar ranges of ϵ parameters are used in prior works such as [132]. After the initial exploration documented in the previous section, our goal now is to focus in on the most accurate and scalable hierarchical methods. Therefore, we omit all flat methods and consider only those values of B that provided satisfactory accuracy. We choose TreeOUECI as our mechanism to instantiate HH (henceforth denoted by HH^c_B, where the c denotes that consistency is applied) method due to its accuracy. We do note that a deployment may prefer TreeHRRCI over TreeOUECI since it requires vastly reduced communication for each user at the cost of only a slight increase in error.

(a) $D = 2^8$

ϵ	HH_2^c	HH_4^c	HH_{16}^c	HaarHRR
0.2	4.269	4.037	4.176	**3.684**
0.4	2.024	2.193	2.590	**1.831**
0.6	1.388	1.341	1.535	**1.278**
0.8	1.002	**0.950**	1.130	0.987
1.0	0.844	**0.744**	0.844	0.811
1.1	0.722	**0.667**	0.820	0.748
1.2	0.684	0.658	**0.642**	0.732
1.4	0.571	**0.542**	0.592	0.601

(b) $D = 2^{16}$

ϵ	HH_2^c	HH_4^c	HH_{16}^c	HaarHRR
0.2	6.745	7.129	8.692	**6.666**
0.4	3.616	**3.424**	4.648	3.526
0.6	**2.333**	2.360	2.793	2.342
0.8	**1.644**	1.728	2.075	1.711
1.0	**1.356**	1.377	1.642	1.484
1.1	1.303	**1.270**	1.597	1.345
1.2	**1.090**	1.140	1.433	1.201
1.4	**0.922**	0.995	1.158	1.130

(c) $D = 2^{20}$

ϵ	HH_2^c	HH_4^c	HH_{16}^c	HaarHRR
0.2	10.043	10.493	11.511	**9.285**
0.4	5.378	**4.751**	5.617	5.261
0.6	3.605	**3.603**	4.483	3.693
0.8	3.047	**3.042**	3.352	3.316
1.0	**2.522**	2.690	3.131	2.915
1.1	2.556	**2.540**	2.729	2.722
1.2	2.619	**2.488**	2.757	2.640
1.4	2.339	**2.304**	2.652	2.505

(d) $D = 2^{22}$

ϵ	HH_2^c	HH_4^c	HaarHRR
0.2	8.629	8.889	**8.422**
0.4	4.546	4.951	**4.470**
0.6	3.181	3.420	**3.085**
0.8	2.657	2.692	**2.462**
1.0	**2.247**	2.358	2.254
1.1	**1.979**	2.252	2.139
1.2	2.120	2.066	**1.946**
1.4	**1.650**	1.885	1.990

Table 5.2: Impact of varying ϵ on mean squared error for arbitrary queries. These numbers are scaled up by 1000 for presentation.

Plot description. Table 5.2 compares the mean squared error for HH_2^c, HH_4^c HH_{16}^c and HaarHRR for various ϵ values. We multiply all results by a factor of 1000 for convenience, so the typical values are around 10^{-3} corresponding to very low absolute error. In each row, we mark in bold the lowest observed variance, noting that in many cases, the "runner-up" is very close behind.

Observations. The first observation, consistent with Figure 5.3, is that for lower ϵ's, HaarHRR is more accurate than the best of HH_B^c methods. This improvement is most pronounced for $D = 2^8$ i.e. at most 10% (at $\epsilon = 0.2$) and marginal (0.01 to 1%) for larger domains. For larger ϵ regimes, HH_B^c outperforms HaarHRR, but only by a small margin of at most 11%. For large domains, HH_B^c remains the best method. In general, except for $D = 2^{22}$, there is no one value of B that achieves the best results at all parameters but overall $B = 4$ yields slightly more accurate results for HH_B^c for most cases. Note that this B value is closer to the optimal value of 9 (derived in Section 5.4.2) than other values. When

(a) $D = 2^8$

ϵ	HH_2^c	HH_4^c	HH_{16}^c	HaarHRR
0.2	4.306	2.968	4.282	**2.857**
0.4	1.859	1.439	1.828	**1.377**
0.6	1.366	**0.957**	1.758	1.031
0.8	0.937	0.778	0.896	**0.758**
1.0	0.802	**0.561**	0.637	0.613
1.1	0.684	**0.533**	0.666	0.626
1.2	0.658	**0.437**	0.670	0.568
1.4	0.573	**0.420**	0.478	0.494

(b) $D = 2^{16}$

ϵ	HH_2^c	HH_4^c	HH_{16}^c	HaarHRR
0.2	7.701	6.172	7.014	**5.870**
0.4	3.266	3.101	3.744	**2.880**
0.6	2.402	2.176	2.426	**2.018**
0.8	1.663	**1.503**	1.834	1.511
1.0	1.338	**1.220**	1.426	1.244
1.1	1.202	**1.051**	1.259	1.120
1.2	1.080	**0.978**	1.147	1.054
1.4	0.973	**0.848**	0.981	0.973

(c) $D = 2^{20}$

ϵ	HH_2^c	HH_4^c	HH_{16}^c	HaarHRR
0.2	8.874	8.255	10.462	**7.237**
0.4	4.734	4.395	5.754	**4.271**
0.6	3.788	3.485	4.055	**3.377**
0.8	3.287	**3.094**	3.268	3.108
1.0	3.022	2.848	**2.826**	2.920
1.1	3.053	2.756	**2.727**	**2.727**
1.2	3.145	**2.627**	2.914	2.754
1.4	2.975	2.659	**2.543**	2.696

(d) $D = 2^{22}$

ϵ	HH_2^c	HH_4^c	HaarHRR
0.2	8.620	8.638	**8.099**
0.4	**4.181**	4.330	4.233
0.6	**2.932**	3.077	3.063
0.8	**2.215**	2.590	2.528
1.0	**1.958**	2.246	2.326
1.1	**1.777**	2.319	2.181
1.2	**1.929**	2.174	2.205
1.4	**1.613**	1.868	2.156

Table 5.3: Impact of varying ϵ on mean squared for prefix queries. These numbers are scaled up by 1000 for presentation. We underline the scores that are smaller than corresponding scores in Table 5.2.

$D = 2^{22}$, HH_2^c dominates HH_4^c but only by a margin of at most 10%.

Comparison with DHT and HH based approaches in the centralized case. We briefly contrast with the conclusion in the centralized case. We reproduce some of the results of Qardaji *et al.* [119] in Table 5.4, comparing variance for the (centralized) wavelet based approach to (centralized) hierarchical histogram approaches with $B = 2, 16$ with consistency applied. These numbers are scaled and not normalized, so can't be directly compared to our results (although, we know that the error should be much lower in the centralized case). However, we can meaningfully compare the *ratio* of variances, which we show in the last two rows of the table.

For $\epsilon = 1, D = 2^8$, the error for the Haar method is approximately 2.8 times more than the hierarchical approach. Meanwhile, the corresponding readings for HaarHRR and HH_4^c (the most accurate method in the $\epsilon = 1$ row) in Table 5.2 are 0.787 and 0.763 — a

D	2^8	2^9	2^{10}	2^{11}
Wavelet	221.62	306.31	410.29	536.32
(optimal) HH^c_{16}	79.23	164.48	185.94	213.87
HH^c_2	220.06	305.54	409.48	535.63
$\frac{\text{Wavelet}}{\text{HH}^c_{16}}$	2.7971	1.8622	2.20	2.5077
$\frac{\text{HH}^c_2}{\text{HH}^c_{16}}$	2.777	1.8576	2.202	2.5044

Table 5.4: Table 3 from [119] comparing the exact average variance incurred in answering all range queries for $\epsilon = 1$ in the centralized case.

deviation of only $\approx$3%. Another important distinction from the centralized case is that we are not penalized a lot for choosing a sub-optimal branching factor. Whereas, we see in the 4th row that choosing $B = 2$ increases the error of consistent HH method by at least 1.8576 times from the preferred method HH^c_{16}.

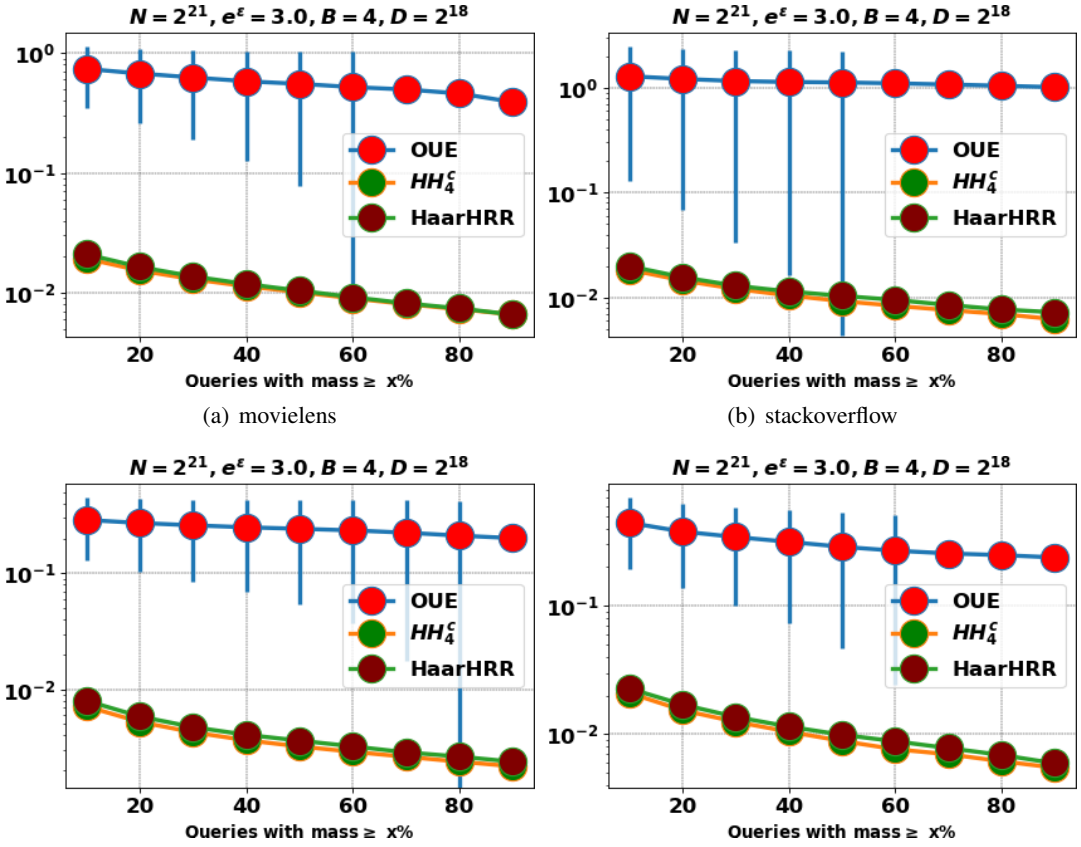
A further observation is that (apart for $D = 2^{22}$) across 24 observations, HaarHRR is never outperformed by *all* values of HH^c_B i.e. in no instance is it the least accurate method. It trails the best HH^c_B method by at most 10%. On the other hand, in the centralized case (Table 5.4), the variance for the wavelet based approach is at least 1.86 times higher than HH^c_B.

5.5.3 Prefix Queries

Experiment description. As described in Section 5.4.4, prefix queries deserve special attention. Our set up is the same as for range queries. We evaluate every prefix query, as there are fewer of them.

Plot description. Table 5.3 is the analogue of Table 5.2 for prefix queries, computed with the same settings. We underline the scores that are smaller than corresponding scores in Table 5.2.

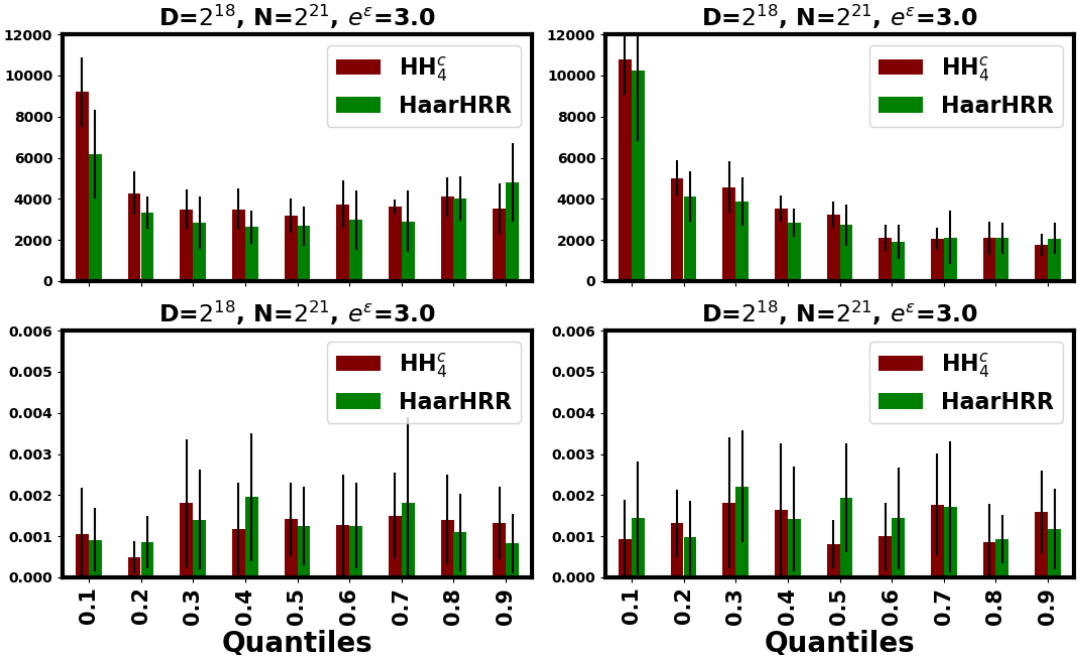
Observations. The first observation is that the error in Table 5.3 is often smaller (up to 30%) than in Table 5.2 at many instances, particularly for small and medium sized domains. The reduction is not as sharp as the analysis might suggest, since that only gives upper bounds on the variance. Reductions in error are not as noticeable for larger values of D, although this could be impacted by our range query sampling strategy. In terms of which method is preferred, HH^c_2 for $D = 2^{22}$ and HH^c_4 tend to dominate for larger ϵ, while HaarHRR is preferred for smaller ϵ.



(a) movielens (b) stackoverflow

(c) NYC yellow cabs data (d) Gowalla check-ins

Figure 5.4: Mean relative error on log scale.



(a) movielens (b) stackoverflow

Figure 5.5: Top row: value error; bottom row: quantile error.

5.5.4 Heavy Intervals

Experiment description. We test the sensitivity of our best hierarchical methods to the heaviness of intervals, i.e. we check whether "heavy hitter" range queries can be answered more accurately than relatively lighter weight queries. In this experiment, we measure error by computing the relative error ($|R - \widehat{R}|/R$) instead of MSE.

Plot description. In each subplot of Figure 5.4, we show the mean relative error for those queries with mass at least x% on log scale. The X axis varies the threshold x from 10 to 90. We include the flat method also for comparison.

Observations. Once again we confirm that the flat method is outperformed by the hierarchical methods even on a different metric by a large margin. For example, in the movielens dataset, the hierarchical methods answer all reasonably heavy queries ($x \geq 10\%$) with $\leq 2\%$ error. The main finding from this figure is that in all datasets, the relative error tends to decrease as x increases. This is to be expected, since the absolute error per query is relatively constant, and so the relative error decreases as the true weight increases.

5.5.5 Quantile Queries

Experiment description. Finally, we compare the performance of the best hierarchical approaches in evaluation of the deciles (i.e. the ϕ-quantiles for ϕ in 0.1 to 0.9) for two real datasets.

Plot description. The top row in Figure 5.5 plots the actual difference between true and reconstructed quantile values (value error). The corresponding bottom plots measure the absolute difference between the quantile value of the returned value and the target quantile (quantile error).

Observations. The first observation is that the both the algorithms have low absolute value error (the top row). For the domain of $2^{18} \approx 262K$, even the largest error of ≈15K made by HH_4^c is still very small, and less than 6%. The value error tends to be the highest where the data is least dense: towards both the extremes for the movielens dataset and only towards the left end for stackoverflow dataset. Importantly, the corresponding quantile error is mostly flat. This means that instead of finding the median (say), our methods return a value that corresponds to the 0.5002 and 0.5003 quantile, which are very close in the distributional sense. This reassures us that any spikes in the value error are mostly a function of sparse data, rather than problems with the methods.

Chapter 6

Count Queries

6.1 Chapter Outline And Our Contributions

In this chapter (based on [41, 42]), we revisit a simple problem — privately releasing the sum of individual's input who satsify a certain property.

• We recall our problem statement, model and present terminologies introduced earlier in more detail in Section 6.2.

• After recollecting the linear programming framework in Section 6.3, we present degeneracies observed in it. In Section 6.4, we present constraints that can be added to avoid these degeneracies.

• In Section 6.5, we revisit the one user/one bit case (Local Differential Privacy), and show that Randomized Response represents a natural convergence of multiple different approaches to privacy.

• In Section 6.6, we observe that some existing approaches yield seemingly undesirable results for small groups (with 1 or a few members), which motivates our further study of differentially private mechanisms. Additional properties which constrain the output can be obtained efficiently via solving a constrained optimization problem. We also propose an explicit construction of a mechanism which provably achieves all our proposed properties, and analyze the additional "cost" in terms of various measures of accuracy.

• Section 6.7 reports on our experiments on accuracy with synthetic and real data.

• We conclude this chapter by extending Ghosh *et al.*'s framework to LDP for histogram aggregation in Section 6.8.

6.2 Model And Preliminaries

The problem of count queries is yet another fundamental problem in private data release that underpins many applications including SQL COUNT * queries. Our model assumes a group of n participants, each of whom has some private information which is encoded as a single bit. They share their information with a trusted aggregator, whose aim is to release information about the sum of the values while protecting the privacy of each participant. We simplify the description of the input to just record the true sum of values j, so we have $0 \leq j \leq n$ to capture the case of a count-query over a table. Our goal is to design a ϵ-DP compliant mechanism $\mathcal{M}$ that, given input j produces output i, subject to certain constraints.

Our mechanism maps inputs in the range 0 to n to outputs in the same range. While one could allow a different set of outputs, it is most natural to restrict to this range. Consider for example, a downstream analysis step which expects counts to be integers in the range $[n]$: we should ensure that this expectation is met by the result of applying mechanisms. Rather than attempt to map different outputs to this range, it is more direct to build mechanisms that cover this output set. It is therefore natural to represent $\mathcal{M}$ as an $(n+1) \times (n+1)$ square matrix $\mathcal{P}$, where $\mathcal{P}_{i,j} = \Pr[\mathcal{M}(j) = i] = \Pr_{\mathcal{M}}[i|j]$. For brevity, we abbreviate this probability to $\Pr[i|j]$. Note that therefore $\mathcal{P}$ is a *column stochastic matrix*: the entries in each column can be interpreted as probabilities, and sum to 1.

We now rephrase the definition of differential privacy in the context of count queries parameterized by $\alpha = \exp(\frac{-\epsilon}{\Delta_1})$. We adopt the α notation for conciseness and retain consistency with the previous works. For count queries, the global sensitivity $\Delta_1 = 1$.

Definition 20 (Differentially Private Mechanisms)**.** *Mechanism $\mathcal{M}$ is α-differentially private for $\alpha \in [0,1]$ if*

$$\forall i, j : \alpha \leq \frac{\Pr[i|j]}{\Pr[i|j+1]} \leq \frac{1}{\alpha}.$$

Here α close to 1 provides a stronger notion of privacy and a tighter constraint on the probabilities, while α close to zero relaxes these constraints. We know that a $\epsilon-$DP mechanism $\mathcal{M}$ with $\Delta_1 = 1$ satisfies (ϵ,1)-LLDP. We say that a DP constraint is *tight* if the relevant inequality is met with equality.

Utility of a mechanism. The true test of the utility of a mechanism is the accuracy with which it allows queries to be answered over real data. However, we aim to design mechanisms prior to their application to data, and so we seek a suitable function to evaluate their quality. Since there are many column stochastic matrices that satisfy DP, the problem of finding a mechanism that provides the maximal utility can be framed as an optimization problem. Specifically, we can encode our notion of utility as a penalty function, where we seek to penalize the mechanism for reporting results that are far from the true answer.

Definition 21 (Objective function value)**.** *We define the objective function $O_{t,\oplus}(\mathcal{P})$ of a mechanism $\mathcal{P}$ as:*

$$O_{t,\oplus}(\mathcal{P}) = \oplus_j \sum_i w_j \Pr[i|j]|i-j|^t$$

where $\oplus$ is an operator like $\sum$ or $\max$, and $\sum_j w_j = 1$.

Observe that the weights w_j can be thought of as a prior distribution on the input values j. Then $O_{t,\sum}(\mathcal{P})$ gives the expected error of the mechanism, when taking its output as the true answer, and $|i-j|^t$ penalizes the extent by which the output was incorrect. When not otherwise stated, we take $w_j = \frac{1}{n+1}$, i.e. a uniform prior over the inputs. Common choices for t in the definition would be $t = 2$, corresponding to a squared error ($\mathbb{L}_2$ norm), $t = 1$, corresponding to an absolute error ($\mathbb{L}_1$ norm), and $t = 0$, corresponding to the probability of any wrong answer ($\mathbb{L}_0$ norm). In what follows, we devote most of our attention to the case $\mathbb{L}_0$. We argue that this is an important case: (i) maximizing the probability of reporting the truth is a natural objective in mechanism design; we aim to ensure that the reported answer is the maximum likelihood estimator (MLE) for the true answer, for use in downstream processing (ii) due to the differential privacy constraints, maximizing the probability of the true answer has the additional effect of making nearby answers likely, as our experiments validate. (iii) our internal study shows that objectives like $\mathbb{L}_1$ and $\mathbb{L}_2$ often give pathological results, as seen in Figure 6.1. Working with $\mathbb{L}_0$ gives more robust behaviour. We therefore initiate the study of constrained mechanism design for $\mathbb{L}_0$, and give some initial results for other objectives. It is convenient to apply a rescaling of the loss function by a factor of $\frac{n+1}{n}$: this sets the cost of a trivial mechanism to 1 (Definition 23). We refer to this rescaled cost as $\mathbb{L}_0$, as this corresponds to a scaled version of $O_{0,\sum}$ that sums the probabilities of a wrong answer, and so

$$\mathbb{L}_0(\mathcal{P}) = \frac{n+1}{n} - \frac{\operatorname{trace}\mathcal{P}}{n}. \tag{6.1}$$

Abusing notation slightly, we also define the objective function, $\mathbb{L}_{0,d} = \frac{n+1}{n}\sum_{i,j:|i-j|\geq d}^{n} w_j \Pr[i|j]$ which computes a rescaled sum of probabilities more than d steps off the main diagonal, so that $\mathbb{L}_0 = \mathbb{L}_{0,0}$.

We have observed following following deficiencies in existing mechanisms.

Defining sampling probabilities: Exponential Mechanism. The framework of exponential mechanism [51] allows us to design mechanisms by specifying a quality function Q mapping input output pairs to real valued scores. This mechanism encodes our preference for providing an output for an input. However, although we can use Q to indicate that some outputs are more desired than others, it is not possible to modify a given Q to directly enforce the properties that we desire, such as ensuring that the probability of returning the true output

is at least as good as that of a uniform distribution ("weak honesty", (6.8)).

Rounding numeric outputs: Laplace and Geometric Mechanisms. Perhaps the best known differentially private mechanism is the Laplace mechanism, which operates by adding random noise to the true answer from an appropriately scaled Laplace distribution. Note that in order to restrict the output range to $[n]$, it will be necessary to round and truncate the output of the mechanism to the range $[n]$. Here, the Laplace mechanism does not easily fit the requirements. Instead, the appropriate method is the discrete analog of the Laplace mechanism, which is the (truncated) Geometric mechanism, introduced by Ghosh *et al.* [36], who showed that it is the basis for unconstrained mechanisms.

Definition 22. *Range Restricted Geometric Mechanism [36] (GM) Let q be the true (unperturbed) result of a count query. The GM responds with $\min(\max(0, q + \delta), n)$, where δ is a noise drawn from a random variable X with a double sided geometric distribution, $\Pr[X = \delta] = \frac{(1-\alpha)^{|\delta|}}{1+\alpha}$ for $\delta \in \mathbb{Z}$.*

That is, GM adds noise from a two sided geometric distribution to the query result and remaps all outputs less than 0 onto 0 and greater than n to n. Though GM does not include rows with all entries zero, we observe that each column distribution in GM has spikes at the extreme values, which tend to distort the true distribution quite dramatically, as the next example shows.

Example 1. *Consider the case of $n = 2$, corresponding to a group of two individuals, with a moderate setting of the privacy parameter $\alpha = \frac{9}{10}$. For an input of 1 (i.e. one user has a 1, and the other has a 0), we obtain that the probability of seeing an output of 0 is ≈ 0.47, and the same for an output of 2. Meanwhile, the probability of reporting the true output is ≈ 0.05 — in other words, the chance of seeing the true answer is eighteen times lower than seeing an incorrect answer. Meanwhile, if the input is 0, then output 0 is returned with probability ≈ 0.53: so the mechanism is much more likely to report the true answer when it is 0 than when it is 1. As we increase the privacy parameter α closer to 1 (more privacy), the probability of outputs other than 0 and n approaches 0.*

As observed in Example 1, an apparent weakness of GM for interpretability is that it can give quite low probabilities for reporting accurate answers. In order to allow more sense to be made of the outputs of the designed mechanisms, we can specify additional constraints to guide the optimization to producing the best interpretable result. This prompts us to define a collection of plausible properties that a mechanism can obey. We will show analytically and empirically that these constraints do not significantly affect the obtained objective function values (i.e. the raw utility), but considerably improve the interpretability of the resulting mechanism. In particular, we demonstrate that it is possible to find a mechanism which achieves all the given properties with only marginal increase in objective function value, and improved interpretability.

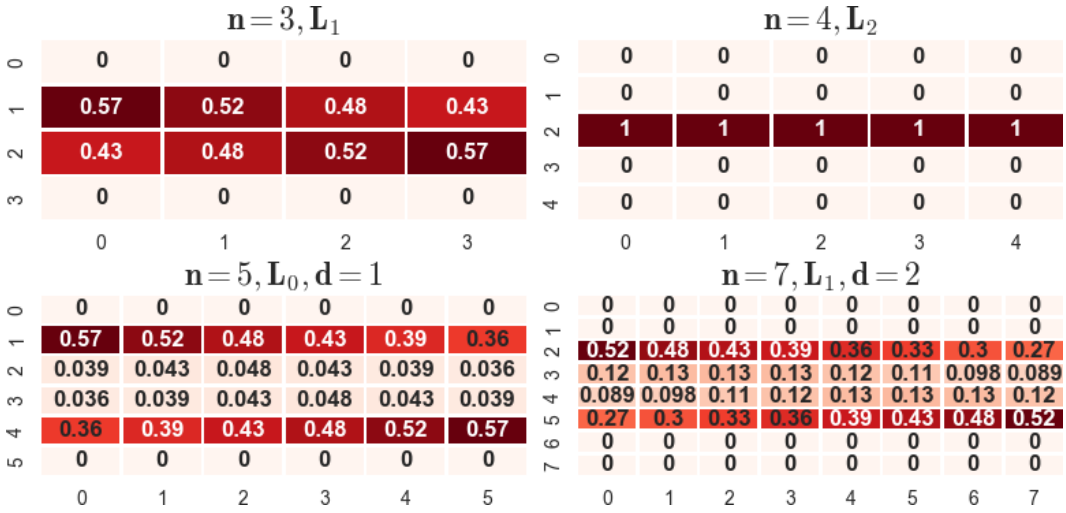


Figure 6.1: Heatmaps of unconstrained mechanisms for $\alpha = 0.62$.

6.3 Unconstrained Mechanism Design

Let's recall Ghosh *et al.*'s formulation from Section 2.4.3 that models the interplay between utility and privacy as a linear program.

$$\begin{aligned} \text{minimize:} \quad & \sum_{j=0}^{n} w_j \sum_{i=0}^{n} |i-j|^p \rho_{i,j} \\ \text{subject to:} \quad & 0 \leq \rho_{i,j} \leq 1 \quad \forall i, j \in [n] \\ & \sum_{i=0}^{n} \rho_{i,j} = 1 \quad \forall j \in [n] \\ \rho_{i,j} \geq \alpha \rho_{i,j+1}, \text{ and } & \rho_{i,j+1} \geq \alpha \rho_{i,j} \quad \forall i \in [n], j \in [n-1] \end{aligned}$$

We refer to the mechanisms obtained from these set of constraints as BASICDP. The result is a linear program with a quadratic number of variables, and a quadratic number of constraints, each containing at most a linear number of variables. Therefore, solving the resulting LP obtains a mechanism minimizing the given objective function with the desired properties, in time polynomial in n.

Anomalies observed in the LP framwork. We found that the "optimal" mechanisms obtained from Ghosh *et al.* 's framework have some anomalous behaviour, such as never reporting some values. Figure 6.1 gives some examples of this phenomenon in action. We show four optimal mechanisms generated by solving linear program described before for different input sizes (n) and loss functions ($\mathbb{L}_0$,$\mathbb{L}_1$ and $\mathbb{L}_2$), under fixed privacy parameter α. Each column gives the probability distribution over the outputs in the range 0 to n, for a given input count (also 0 to n). The case of optimizing the squared error ($\mathbb{L}_2$) is most striking:

the "optimal" thing to do in this case is to ignore the input and always report '2'! But other cases are also problematic: all these optimal mechanisms never report some outputs (gaps), and disproportionately report some others (spikes). For example, minimizing the absolute error for $n = 7$ has a chance of reporting the values 2 or 5 with at least 0.7 probability, regardless of the input value. Similarly, if we try to minimize the probability of reporting an answer that is more than 1 step away from the true input (denoted as $\mathbb{L}_0$ with $d = 1$), there is an over 90% chance of reporting 1 or 4. Our studies found that similar undesirable results were found across a range of choices of n, α and loss function. Simple attempts to prevent these outcomes are not effective. For example, we can ensure that no entry is zero by adding a constraint to the LP enforcing this. However, the consequence is that rows which were zero are now set to be the smallest allowable value, which is unsatisfying. Instead, we propose an additional set of properties to eliminate degeneracy and provide more structure in our solutions.

6.4 Constrained Mechanism Design

We now propose a set of structural properties that help to control the objective function in addition to meeting differential privacy. We believe that these constraints are natural and intuitive and often observed in other mechanisms satisfying differential privacy. We present properties of three types: those which operate on rows of the matrix, those which apply to columns of the matrix, and those which apply to the diagonal.

Row Honesty (RH). A mechanism is *row honest* if

$$\forall i, j. \Pr[i|i] \geq \Pr[i|j] \tag{6.2}$$

Row honesty means that a mechanism should have higher probability of reporting i when the input is i than for any other input.

Row Monotone (RM). A mechanism is *row monotone* if

$$\begin{aligned} &\forall 1 \leq j \leq i : \Pr[i|j-1] \leq \Pr[i|j] \\ &\forall i \leq j < n : \Pr[i|j+1] \leq \Pr[i|j] \end{aligned} \tag{6.3}$$

This property generalizes row honesty: row monotonicity implies row honesty. It requires that entries in row i are monotone non-increasing as we move away from the diagonal element $\Pr[i|i]$. Note that row monotonicity is independent of differential privacy: we can find mechanisms that achieve DP but are not row monotone, and vice-versa.

Analogous to the row-wise properties, we define monotonicity and honesty along columns also.

Column Honesty (CH). A mechanism is *column honest* if

$$\forall i, j : \Pr[j|j] \geq \Pr[i|j]. \tag{6.4}$$

Column honesty requires that the mechanism be *honest* enough to report the true answer more often than any individual false answer. As demonstrated by Example 1, GM does not obey column honesty.

Column Monotone (CM). A mechanism is *column monotone* if

$$\begin{aligned} \forall 1 \leq i \leq j : \Pr[i-1|j] \leq \Pr[i|j] \\ \forall j \leq i < n : \Pr[i+1|j] \leq \Pr[i|j] \end{aligned} \tag{6.5}$$

As in the row-wise case, column monotonicity implies column honesty (but not vice-versa). It captures the property that outputs closer to the true answer should be more likely than those further away.

Fairness (F). A mechanism is *fair* when the probability of reporting the true input is constant, i.e.

$$\forall i, j : \Pr[i|i] = \Pr[j|j] := y. \tag{6.6}$$

Example 1 shows that GM is not a fair mechanism. If a mechanism is fair and has row honesty, then all off-diagonal elements are at most y, so the mechanism also satisfies column honesty. Symmetrically, a fair and column honest mechanism is row honest. While this may seem like a restrictive constraint, we observe that mechanisms proposed in other contexts have this property, such as the staircase mechanism of [140].

Lemma 11. *If a mechanism is required to be fair, then any mechanism that minimizes the objective $O_{0,\sum}$ is simultaneously optimal for all settings of weights w_j.*

Proof. Let the diagonal element of the fair mechanism be y. The objective function value is

$$\sum_{j \in [n]} \sum_{i \in [n]} w_j \Pr[i|j](i-j)^0 = \sum_{j \in [n]} w_j(1-y) = 1 - y \tag{6.7}$$

That is, the value is independent of the w_js. □

Weak Honesty (WH). A mechanism satisfies *weak honesty* if

$$\forall i : \Pr[i|i] \geq \frac{1}{n+1} \tag{6.8}$$

We can consider this property a weaker version of column honesty, as CH implies WH: for

any column j, summing the column honesty property over all rows i we obtain

$$(n+1)\Pr[i|i] = \sum_{i=0}^{n} \Pr[j|j] \geq \sum_{i=0}^{n} \Pr[i|j] = 1$$

so after rearranging, we have $\Pr[i|i] \geq \frac{1}{n+1}$. Weak honesty ensures that a mechanism reports the true answer with probability at least that of uniform guessing (formalized as the uniform mechanism UM in Definition 23). It also ensures that the mechanism does not have any rows that are all zero (corresponding to outputs with no probability of being produced). GM does not always obey weak honesty, as is shown by Example 1.

The final property we consider is a natural symmetry property (formally, it is that the matrix $\mathcal{P}$ is *centrosymmetric*):

Symmetry (S). A mechanism is *symmetric* if

$$\forall i, j : \Pr[i|j] = \Pr[n-i|n-j] \tag{6.9}$$

Since the input and output domains, and the objective functions are symmetric, it is natural to seek mechanisms which are also symmetric. Our next result shows that symmetry is always achievable without any loss in objective function.

Theorem 8. *Given a mechanism M which meets a subset of properties P from those defined above, we can construct a symmetric mechanism M^* which also satisfies all of P and achieves the same objective function value as M.*

Proof. Our construction to achieve symmetry is simple. Define a matrix M^S from M as $(M^S)_{i,j} = M_{n-i,n-j}$. Then set $M^* = \frac{1}{2}(M + M^S)$. We first observe that M^* is indeed symmetric, since $M^*_{i,j}$ is equal to

$$\tfrac{1}{2}(M_{i,j} + M_{n-i,n-j}) = \tfrac{1}{2}(M_{n-i,n-j} + M_{n-(n-i),n-(n-j)}) = M^*_{n-i,n-j}$$

as required by (6.9). The ($\mathbb{L}_0$) objective function value is unchanged since (invoking (6.1))

$$\operatorname{trace}(M^*) = \frac{1}{2}(\operatorname{trace}(M) + \operatorname{trace}(M^S)) = \operatorname{trace}(M)$$

For the other diagonal properties (fairness and weak honesty), it is immediate that if either of these properties are satisfied by M, then they are also satisfied by M^*. We prove the claim for row properties; the case for column properties is symmetric.

(i) Differential privacy: if we have $\alpha \leq M_{i,j}/M_{i,j+1} \leq 1/\alpha$ for all i, j, then this also holds for $M^S_{i,j}/M^S_{i,j+1}$. Summing both inequalities, and using that $\min(\frac{a}{b}, \frac{c}{d}) \leq \frac{a+c}{b+d} \leq \max(\frac{a}{b}, \frac{c}{d})$, this holds for M^*, hence M^* satisfies differential privacy.

(ii) Row monotonicity: consider a pair i, j with $1 \leq i \leq j$. Then we have $M_{j,i-1} \leq M_{j,i}$ (from (6.3)). It is also the case that $n - j \leq n - i < n$, which means that $M_{n-j,n-i+1} \leq$

$M_{n-j,n-i}$ (also from (6.3)). Then $M^S_{j,i-1} \leq M^S_{j,i}$. Combining these two inequalities, we have that $M^*_{j,i-1} \leq M^*_{j,i}$.
(iii) Row honesty: if $\forall i, j.M_{i,i} \geq M_{i,j}$, then $M^S_{i,i} \geq M^S_{i,j}$ also. Summing both inequalities, we obtain $M^*_{i,i} \geq M^*_{i,j}$ as required. □

Consequences of these properties. We first argue that these properties all contribute to avoiding the degenerate mechanisms shown above. The (column, row) honesty and monotonicity properties work to prevent the "spikes" observed when a value far from the true input is made excessively likely. The (column) honesty properties do so by preventing a far output being more likely than the true input; the (column) monotonicity properties do so more strongly by ensuring that any further output is no more likely than one that is nearer to the true input. Fairness, column honesty and weak honesty prevent gaps (zero rows): they ensure that the diagonal entry in each row is non-zero, and then the DP requirement ensures that all other entries in the same row must also be non-zero. We next show that there is an efficient procedure to find an optimal constrained mechanism for any $n > 1$.

Theorem 9. *Given any subset of the structural constraints, we can find an optimal (constrained) mechanism which respects these constraints in time polynomial in n.*

Proof. We break the proof into two pieces. First, we argue that given any subset of structural constraints we can create a Linear Program describing it, and second we argue that there exists a mechanism satisfying them all. Observe that all seven properties listed above can be encoded as linear constraints. For example, symmetry is written as

$$\rho_{i,j} = \rho_{n-i,n-j} \quad \forall i, j \in [n]$$

while weak honesty is

$$\rho_{i,i} \geq 1/(n+1).$$

Row monotonicity becomes

$$\rho_{j,i-1} \leq \rho_{j,i} \quad \forall j \in [n], i < j$$
$$\rho_{j,i+1} \leq \rho_{j,i} \quad \forall i \in [n-1], j < i$$

Consequently, we can create a linear program of size polynomial in n, by adding these to the BASICDP constraints (2.9), (2.10) and (2.11) established in Section 6.3. This shows the first part of the proof. Next, we show that any such LP is feasible by defining a trivial baseline mechanism:

Definition 23 (Uniform Mechanism, UM)**.** *The* uniform mechanism *of size n has* $\Pr[i|j] = \frac{1}{n+1}$*, for all $i, j \in [n]$.*

That is, UM ignores its input and picks an allowable output uniformly at random. It demonstrates that all our properties are (simultaneously) achievable, albeit trivially. By observation, the mechanism is symmetric and fair for any $\alpha' \leq 1$. It meets the inequalities specified for row monotonicity, column monotonicity and weak honesty with equality. UM also satisfies differential privacy for all $\alpha \leq 1$. □

Clearly, UM is undesirable from the perspective of providing utility. We easily calculate that the objective function value $O_{0,\sum}$ achieved by UM is $\frac{n}{n+1}$, which is close to the maximum possible value of 1. Note that we chose our definition of the $\mathbb{L}_0$ function to assign this mechanism a (reweighted) score of 1.

6.5 Constrained Mechanisms: $n = 1$

In this section, we consider an important special case of our problem: where a single user has a single private bit value. This is the limiting case of our setting, corresponding to $n = 1$. It turns out to be an important scenario that has been studied over many decades, as it asks each user to reveal a (noisy) version of their information for subsequent aggregation. We briefly revisit this case in the light of the objectives and properties defined above. The main conclusions we find are that for $n = 1$, all approaches to building DP mechanisms are essentially the same, and trivially obey all our constraints, making this a starting point for our subsequent study.

6.5.1 Randomized Response (RR)

Theorem 10. *In the one bit (binary) case, RR is the unique optimal non-trivial α-differentially private mechanism under any objective function $O_{t,\sum}$ when $\alpha \leq w_1/w_0 \leq 1/\alpha$.*

Proof. The objective function is to minimize

$$w_0 \Pr[1|0]1^p + w_1 \Pr[0|1]1^p = w_0 \Pr[1|0] + w_1 \Pr[0|1].$$

To achieve α-differential privacy, we must have

$$\Pr[0|0] \leq \frac{1}{\alpha} \Pr[0|1] \qquad \text{and} \qquad \Pr[1|1] \leq \frac{1}{\alpha} \Pr[1|0]. \tag{6.10}$$

We can observe that in order to minimize the objective function (for non-negative w_0 and w_1), it suffices to maximize $\Pr[0|0]$ and $\Pr[1|1]$, and so the inequalities in (6.10) become

equalities. Thus, our objective function to minimize becomes:

$$\begin{aligned} w_0(1 - \Pr[0|0]) + w_1 \Pr[0|1] &= w_0(1 - \frac{1}{\alpha}\Pr[0|1]) + w_1 \Pr[0|1] \\ &= w_0 + (w_1 - \frac{1}{\alpha}w_0)\Pr[0|1]. \end{aligned}$$

In the case that $w_1/w_0 < \frac{1}{\alpha}$ (or, symmetrically, if $w_0/w_1 < \frac{1}{\alpha}$), the trivial solution is $\Pr[0|1] = \Pr[0|0] = 1$, i.e. the mechanism ignores the input and always reports '0' (in the symmetric case, it always reports '1').

Otherwise $\alpha \leq w_1/w_0 \leq 1/\alpha$, and we have

$$\begin{aligned} \Pr[0|0] = \frac{1}{\alpha}\Pr[0|1] &= \frac{1}{\alpha}(1 - \Pr[1|1]) \\ &= \frac{1}{\alpha}(1 - \frac{1}{\alpha}\Pr[1|0]) \\ &= \frac{1}{\alpha}(1 - \frac{1}{\alpha}(1 - \Pr[0|0])) \end{aligned}$$

Rearranging, we obtain

$$\Pr[0|0](1 - \frac{1}{\alpha}^2) = \frac{1}{\alpha}(1 - \frac{1}{\alpha})$$

and so $\Pr[0|0] = \frac{1}{1+\alpha}$, and $\Pr[1|1] = 1 - \Pr[0|0]/\alpha = \frac{1}{1+\alpha}$.
Consequently, we obtain an instance of randomized response with $p = \frac{1}{1+\alpha}$.
□

Fact 4. *For $p \geq \frac{1}{2}$, RR satisfies all properties listed in Section 6.4*

The fact follows immediately by representing RR as a 2×2 matrix below and visually inspecting.

$$\mathcal{R} = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}$$

This entails fairness and symmetry. All other properties reduce to the condition that $p \geq 1-p$, i.e. $p \geq \frac{1}{2}$.

6.5.2 Exponential Mechanism

Theorem 11. *In the one bit (binary) case, the Exponential Mechanism results in an instance of Randomized Response with $p = \frac{\exp(\epsilon/2)}{1+\exp(\epsilon/2)}$.*

Proof. In the binary case, we have $D = R = \{0, 1\}$. Without loss of generality, we can assume that $Q(0, 0) = Q(1, 1) := c$; $Q(1, 0) = Q(0, 1) := w$ (if not, this makes the privacy

$$\begin{pmatrix} x & x\alpha & x\alpha^2 & x\alpha^3 & \cdots & x\alpha^n \\ y\alpha & y & y\alpha & y\alpha^2 & \cdots & y\alpha^{n-1} \\ y\alpha^2 & y\alpha & y & y\alpha & \cdots & y\alpha^{n-2} \\ y\alpha^3 & y\alpha^2 & y\alpha & y & \cdots & y\alpha^{n-3} \\ y\alpha^4 & y\alpha^3 & y\alpha^2 & y\alpha & \cdots & y\alpha^{n-4} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x\alpha^n & x\alpha^{n-1} & x\alpha^{n-2} & x\alpha^{n-3} & \cdots & x \end{pmatrix}$$

Figure 6.2: Structure of GM, where $x = \frac{1}{1+\alpha}$ and $y = \frac{1-\alpha}{1+\alpha}$.

guarantee loose in one case). We also assume that $c \geq w$, since we should make the true response more likely than the incorrect response. Then, by definition, $s = c - w$. The resulting mechanism has

$$\begin{aligned} \Pr[0|0] =& \frac{\exp(\epsilon c/2s)}{\exp(\epsilon w/2s) + \exp(\epsilon c/2s)} \\ =& \frac{\exp(-w\epsilon/2s)}{\exp(-w\epsilon/2s)} \frac{\exp(\epsilon c/2s)}{\exp(\epsilon w/2s) + \exp(\epsilon c/2s)} \\ =& \frac{\exp(\epsilon/2)}{1 + \exp(\epsilon/2)} \end{aligned}$$

Meanwhile, $\Pr[1|0] = 1 - \Pr[0|0] = 1/(1 + \exp(\epsilon/2))$, $\Pr[1|1] = \Pr[0|0]$ and $\Pr[0|1] = \Pr[1|0]$. Consequently, the mechanism is equivalent to $\mathcal{R}$ from Fact 4, and the privacy guarantee is given by $\Pr[0|0]/\Pr[1|0] = \exp(-\epsilon/2)$. □

Note that this direct application of the exponential mechanism construction actually yields $\exp(-\epsilon/2)$ privacy, stronger than specified, since it does not take full advantage of the additional simple structure of this scenario.

6.5.3 Geometric Mechanism

Lemma 12. *In the binary case, the Geometric mechanism results in an instance of Randomized Response with* $p = \frac{1}{1+\alpha}$.

Proof. When $n = 1$, we can consider each input separately. On input 0, the output is 0 if $\delta \leq 0$. From properties of the geometric distribution, we obtain

$$\Pr[0|0] = \Pr[X \leq 0] = \frac{1-\alpha}{1+\alpha}.(1 + \alpha + \alpha^2 + \alpha^3 + \ldots) = \frac{1}{1+\alpha}.$$

Then, $\Pr[1|0] = \Pr[X > 0] = \frac{\alpha}{1+\alpha}$. The case for input 1 is symmetric. Hence the claim follows. □

6.6 Constrained mechanisms: $n > 1$

For $n > 1$, it is not the case that all mechanisms automatically achieve all our enumerated properties. In this section, we consider mechanisms achieving various combinations of the structural properties.

6.6.1 The Geometric Mechanism

Next, we return to the GM (Definition 22). In Figure 6.2, we show the structure of the mechanism, which can be derived by simple calculation from Definition 22. Below, we show that it enjoys a number of special properties. In prior work, Ghosh *et al.* showed that GM plays an important role, as it can be transformed into an optimal mechanism for different objectives. Here, we argue a more direct result: that GM is directly optimal for a uniform objective function[1]

Theorem 12. *GM is the (unique) optimal mechanism satisfying* BASICDP *under the* $\mathbb{L}_0$ *objective function.*

Proof. In order to prove the theorem, we define a modified form of a mechanism which is row monotone and in which all the DP inequalities are tight. Given a mechanism $\mathcal{P}$ whose leading diagonal is $y = [y_0, y_1, \dots y_n]$, define $\mathcal{P}'$ as the unique row monotone matrix where all the DP inequalities are tight. That is,

$$\mathcal{P}' = \begin{pmatrix} \mathbf{y_0} & y_0\alpha & y_0\alpha^2 & y_0\alpha^3 & \cdots & y_0\alpha^n \\ y_1\alpha & \mathbf{y_1} & y_1\alpha & y_1\alpha^2 & \cdots & y_1\alpha^{n-1} \\ y_2\alpha^2 & y_2\alpha & \mathbf{y_2} & y_2\alpha & \cdots & y_2\alpha^{n-2} \\ y_3\alpha^3 & y_3\alpha^2 & y_3\alpha & \mathbf{y_3} & \cdots & y_3\alpha^{n-3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ y_n\alpha^n & y_n\alpha^{n-1} & y_n\alpha^{n-2} & y_n\alpha^{n-3} & \cdots & \mathbf{y_n} \end{pmatrix}$$

Note that $\mathcal{P}'$ is dominated by $\mathcal{P}$, in the sense that $\mathcal{P}'_{i,j} \leq \mathcal{P}_{i,j}$ for all i and j. This holds because, given y_i, the DP constraints enforce that $\mathcal{P}_{i,j}$ cannot be less than $y_i\alpha^{|i-j|}$, which is exactly the value of $\mathcal{P}'_{i,j}$. However, $\mathcal{P}$ is not strictly a mechanism, since it is not guaranteed to be column stochastic: columns may sum to less than one. To address this, we define a 'slack vector' s so that $s_j = 1 - \sum_{i=0}^{n} \mathcal{P}'_{i,j}$. In finding an optimal mechanism $\mathcal{P}$, we seek to maximize $\mathrm{trace}(\mathcal{P})$ (from (6.1)). Since $\mathrm{trace}(\mathcal{P}) = \mathrm{trace}(\mathcal{P}')$ by definition, we can concentrate on $\mathcal{P}'$ and seek to maximize its trace. We interpret the slack variables s as "missed potential". Observe that each s_j represents probability mass that could (perhaps)

[1]Note that, compared to [36], we define mechanisms to enforce differential privacy along rows of $\mathcal{P}$ rather than columns.

be added to $\mathcal{P}_{j,j}$ to increase the trace. Therefore, in order to maximize the trace, we seek to minimize the slack. Note that for any given slack vector s and parameter α, there is at most one mechanism $\mathcal{P}'$ whose slack vector is s: there are $n+1$ unknowns y_j, and $n+1$ constraints relating these to s. Specifically, let $A(\alpha)$ be the Toeplitz matrix such that $A(\alpha)_{i,j} = \alpha^{|i-j|}$. Then given α and s, we seek the solution y to $A(\alpha)y = \mathbf{1}_{n+1} - s$, where $\mathbf{1}_{n+1}$ is the $n+1$ length vector whose every entry is 1.

We now show that there exists a feasible solution to this system with $s = 0$, that is with no slack values. In this case, $\mathcal{P} = \mathcal{P}'$ and is optimal as there is no remaining slack potential that could increase the trace. From the first row of $A(\alpha)$, corresponding to the first column of $\mathcal{P}'$, we have

$$y_0 + y_n\alpha^n + \sum_{i=1}^{n-1} y_i\alpha^i = 1 \tag{6.11}$$

Similarly, from the second column of $\mathcal{P}'$,

$$y_0\alpha + y_n\alpha^{n-1} + \sum_{i=1}^{n-1} y_i\alpha^{i-1} = 1 \tag{6.12}$$

$$\text{so } y_0\alpha^2 + y_n\alpha^n + \sum_{i=1}^{n-1} y_i\alpha^i = \alpha \tag{6.13}$$

Then, combining (6.11) and (6.13), we obtain

$$y_0\alpha^2 + y_n\alpha^n + (1 - y_0 - y_n\alpha^n) = \alpha$$

which yields $y_0 = \frac{1}{1+\alpha}$. Following the same approach for columns n and $n+1$ of $\mathcal{P}'$, we similarly obtain $y_n = \frac{1}{1+\alpha}$. We find each remaining y_i in turn, starting from y_1. Taking the linear combination which subtracts α times column $i+1$ of $\mathcal{P}'$ from column i of $\mathcal{P}'$ eliminates $y_{i+1} \dots y_{n-1}$. We then obtain

$$y_0\alpha^i(1-\alpha^2) + (1-\alpha^2)\sum_{j=1}^{i} y_j\alpha^{i-j} = 1 - \alpha.$$

Substituting the found value of y_0, we obtain

$$\frac{\alpha^i}{1+\alpha} + \sum_{j=1}^{i} y_j\alpha^{i-j} = \frac{1}{1+\alpha}$$

$$\sum_{j=1}^{i} y_j\alpha^{i-j} = \frac{1-\alpha^i}{1+\alpha}.$$

The base case $i = 1$ yields $y_1 = \frac{1-\alpha}{1+\alpha}$. Then, inductively, $y_i = \frac{1-\alpha}{1+\alpha}$. Assuming the inductive hypothesis, we have

$$\sum_{j=1}^{i-1} \frac{(1-\alpha)\alpha^j}{1+\alpha} + y_i = \frac{1-\alpha^i}{1+\alpha}.$$

$$\text{Simplifying, } \frac{1-\alpha}{1+\alpha} \sum_{j=0}^{i-1} \alpha^j - \frac{1-\alpha}{1+\alpha} + y_i = \frac{1-\alpha^i}{1+\alpha}.$$

Using the standard expression for the sum of a geometric progression, the summation term becomes $\frac{1-\alpha^i}{1-\alpha}$. Substituting this and cancelling, we find $y_i = \frac{1-\alpha}{1+\alpha}$.

To complete the proof, we observe that the resulting mechanism $\mathcal{P} = \mathcal{P}'$ defined by the diagonal

$$y = \left[\frac{1}{1+\alpha}, \frac{1-\alpha}{1+\alpha}, \frac{1-\alpha}{1+\alpha} \ldots, \frac{1-\alpha}{1+\alpha}, \frac{1}{1+\alpha} \right]$$

is exactly GM, by comparison to Figure 6.2. Hence, the optimal mechanism **OPT** has a unique form, which is GM. □

Limitations of GM. Since GM is 'optimal' for $\mathbb{L}_0$, should we conclude our study here? The answer is no, since GM fails to satisfy many of the desirable properties we identified in Section 6.4, and as illustrated in Example 1. We have already observed that GM is not fair, and does not in general satisfy column honesty (or column monotonicity) or weak honesty. Next, we identify parameter settings for when they do hold.

Lemma 13. *GM obeys weak honesty iff* $n \geq \frac{2\alpha}{1-\alpha}$.

Proof. Weak honesty requires the diagonal elements to all exceed $\frac{1}{n+1}$. Since $y < x$, we focus on y. We require $y \geq \frac{1}{n+1}$ i.e $\frac{1-\alpha}{1+\alpha} \geq \frac{1}{n+1}$. This reduces to $n + 1 \geq \frac{1+\alpha}{1-\alpha}$, giving the requirement $n \geq \frac{2\alpha}{1-\alpha}$. □

GM satisfies the column monotonicity condition for many i, j pairs. The critical place in the matrix where it can be violated is between the first and second rows (symmetrically, between penultimate and final rows). This corresponds to the problematic behaviour of GM to report extreme outputs (0 or n) overly often in the increased privacy regime ($\alpha > \frac{1}{2}$).

Lemma 14. *GM achieves column monotonicity iff* $\alpha \leq \frac{1}{2}$.

Proof. We require $\Pr[1|1] \leq \Pr[0|1]$, i.e. $y \leq \alpha x$ or $\frac{1-\alpha}{1+\alpha} \leq \frac{\alpha}{1+\alpha}$. This gives the condition $\alpha \leq \frac{1}{2}$. It is straightforward to check that this ensures monotonicity in all other columns. □

By inspection, GM is always symmetric, and row monotone. The ($\mathbb{L}_0$) objective function value achieved by GM is

$$\frac{n+1}{n}\big(1-\frac{(n-1)y+2x}{n+1}\big)=\frac{n+1}{n}\big(1-\frac{n-1}{n+1}\frac{1-\alpha}{1+\alpha}-\frac{2}{(1+\alpha)(n+1)}\big)=\frac{2\alpha}{1+\alpha}$$

We next design a different explicit mechanism which achieves more of the desired properties.

6.6.2 Explicit Fair Mechanism

Although we can achieve any desired combination of properties by solving an appropriate linear program, it is natural to ask whether there is any non-trivial explicit mechanism that achieves properties such as fairness with an objective function score comparable to that of GM. We answer this question in the positive. First, we consider the limits of what can be achieved under fairness. In the case of GM, all DP inequalities are tight. This is not possible when fairness is demanded. A fair mechanism M with all DP inequalities tight would be completely determined: $M_{i,j}=y\alpha^{|i-j|}$ for some y. It is easy to calculate for any such mechanism that there is no setting of y which ensures that all columns sum to 1, a contradiction. Hence, we cannot have a fair mechanism with all DP inequalities tight. Nevertheless, trying to achieve tightness provides us with a bound on what can be achieved.

Lemma 15. *Let F be a fair mechanism of size $(n+1)\times(n+1)$ with y as the diagonal element. Then $y\leq\frac{1-\alpha}{1+\alpha-2\alpha^{\frac{n}{2}+1}}$.*

Proof. There are some slight differences depending on whether we consider odd or even values of n. Without loss of generality, take n even. We will consider a fixed column j. For all i, we are required to have $\Pr[i|i]=y$ for some y. Repeatedly applying the DP inequality, we obtain an upper bound involving y as $\Pr[i|j]\geq y\alpha^{i-j}$ when $j<i$ and $\Pr[i|j]\geq y\alpha^{j-i}$ when $i>j$. Summing these for any given column j and equating to 1 provides an upper bound on y. We get the tightest bound by picking column $j=\frac{n}{2}$. Then $y+2y\sum_{j=1}^{\frac{n}{2}}\alpha^j\leq 1$, so:

$$y\leq\frac{1}{1+2\sum_{j=1}^{\frac{n}{2}}\alpha^j}=\frac{1-\alpha}{1+\alpha-2\alpha^{\frac{n}{2}+1}}\tag{6.14}$$

For n large enough, we can neglect the $\alpha^{n/2+1}$ term, and approximate this quantity by $\frac{1-\alpha}{1+\alpha}$. □

Note that for optimality under an objective function $O_{t,\sum}$, we should make y as large as possible. Hence, any optimal mechanism will have y as close to this value as possible. Indeed, the above proof helps us to design an explicit mechanism EM that achieves fairness. The proof argues that in column $n/2$, the smallest values we can obtain above and

$$\begin{pmatrix} y & y\alpha & y\alpha^2 & y\alpha^3 & y\alpha^4 & y\alpha^4 & y\alpha^4 & y\alpha^4 \\ y\alpha & y & y\alpha & y\alpha^2 & y\alpha^3 & y\alpha^3 & y\alpha^3 & y\alpha^3 \\ y\alpha & y\alpha & y & y\alpha & y\alpha^2 & y\alpha^3 & y\alpha^3 & y\alpha^3 \\ y\alpha^2 & y\alpha^2 & y\alpha & y & y\alpha & y\alpha^2 & y\alpha^2 & y\alpha^2 \\ y\alpha^2 & y\alpha^2 & y\alpha^2 & y\alpha & y & y\alpha & y\alpha^2 & y\alpha^2 \\ y\alpha^3 & y\alpha^3 & y\alpha^3 & y\alpha^2 & y\alpha & y & y\alpha & y\alpha \\ y\alpha^3 & y\alpha^3 & y\alpha^3 & y\alpha^3 & y\alpha^2 & y\alpha & y & y\alpha \\ y\alpha^4 & y\alpha^4 & y\alpha^4 & y\alpha^4 & y\alpha^3 & y\alpha^2 & y\alpha & y \end{pmatrix}$$

Figure 6.3: Explicit fair mechanism for $n = 7$.

below the y entry are αy, $\alpha^2 y$ and so on up to $\alpha^{n/2} y$. Then the sum of these terms is set to 1. All other columns must also sum to 1; a simple way to achieve this is to ensure all columns contain a permutation of the same set of terms. To ensure DP is satisfied, we should arrange these so that row-adjacent entries differ in their power of α by at most one. Our explicit fair mechanism EM is then defined as follows:

$$\Pr[i|j] = \begin{cases} y\alpha^{|i-j|} & \text{if } |i-j| < \min(j, n-j) \\ y\alpha^{\lceil \frac{|i-j|+\min(j,n-j)}{2} \rceil} & \text{otherwise} \end{cases} \tag{6.15}$$

Here, y is set to $\frac{1-\alpha}{1+\alpha-2\alpha^{n/2+1}}$, i.e. the value determined in Equation (6.14). From the proof of Lemma 11 and (6.1), we have that the $\mathbb{L}_0$ score of this mechanism is $\frac{n+1}{n}(1-y)$, as it maximizes y subject to the bound of Lemma 11. Figure 6.3 shows the instantiation of this mechanism for the case $n = 7$. Comparing to GM, we see that the diagonal elements are slightly increased, with the exception of the two corner diagonals, which are decreased. It is tempting to try to obtain the mechanism via the Exponential Mechanism, by using a quality function applied to $|i-j|$ similar in form to (6.15). Note however, that the constant factors of 2 in its definition (2.2) leads to a considerably weaker result than this explicit construction, equivalent to halving the privacy parameter ϵ. It is easy to check that in the $n = 7$ example, the mechanism is symmetric, and meets all of the properties defined in Section 6.4. In fact, this is the case for all values of n. The proof is rather lengthy and proceeds by considering a number of cases.

Theorem 13. *EM is an optimal mechanism under $\mathbb{L}_0$ that satisfies all properties listed in Section 6.4.*

Proof. That EM is fair follows by definition: for $\Pr[i|i]$, the definition gives $y\alpha^0 = y$ in all cases. Next, we argue that all column sums are 1, i.e. EM is a valid mechanism. Consider some column $j \leq n$. Observe that fixing j determines which of j and $n-j$ is smaller. Assume that it is j, i.e. $j \leq n/2$ (the other case is symmetric), and assume n is even. Then

we have

$$\sum_{i=0}^{n} \Pr[i|j] = \sum_{|i-j|<j} y\alpha^{|i-j|} + \sum_{|i-j|\geq j} y\alpha^{\lceil \frac{1}{2}(|i-j|+j)\rceil}$$
$$= y + \sum_{i=1}^{j} 2y\alpha^{i} + \sum_{i=j}^{n} y\alpha^{\lceil \frac{1}{2}i\rceil} = y + \sum_{i=1}^{n/2} 2y\alpha^{i}$$

This sums to 1 given our choice of y. For n odd, the calculation is the same except there is one additional term of $y\alpha^{\lceil n/2\rceil}$ in the final sum (and we choose y to ensure that this sum is 1). The mechanism meets our definition of symmetry (6.9), since according to (6.15), $\Pr[n-i|n-j]$ is given by

$$y\alpha^{|(n-i)-(n-j)|} \text{ if } |(n-i)-(n-j)| < \min(n-j, n-(n-j))$$
$$y\alpha^{\lceil \frac{|(n-i)-(n-j)|+\min(n-j,n-(n-j))}{2}\rceil} \text{ otherwise}$$

Simplifying this expression, we observe that it is identical to (6.15).

Column Properties. Consider a fixed column j of the mechanism. As we look at neighboring entries i and $i+1$, we have four cases:

Case (1): $|i-j| < \min(j, n-j)$ and $|i+1-j| < \min(j, n-j)$.
Then $\Pr[i|j] = y\alpha^{|i-j|}$ and $\Pr[i+1|j] = y\alpha^{|i+1-j|}$, so the probability either increases by a factor of α (when $j < i$) or increases by a factor of α (when $j \geq i$).

Case (2): $|i-j| \geq \min(j, n-j)$ and
$|i+1-j| \geq \min(j, n-j)$.
Then $\Pr[i|j] = y\alpha^{\lceil \frac{|i-j|+\min(j,n-j)}{2}\rceil}$, while $\Pr[i+1|j] = y\alpha^{\lceil \frac{|i+1-j|+\min(j,n-j)}{2}\rceil}$. Depending on the parity of i, the latter probability can only stay the same; increase by a factor of α (only when $i > j$); or decrease by a factor of α (only when $j > i$).

Case (3): $|i-j| \geq \min(j, n-j)$ but $|i+1-j| < \min(j, n-j)$.
Then we must have $i < j$ for both conditions to hold. So we must have (combining the two conditions)

$$j - i \geq \min(j, n-j) > j - (i+1)$$

We have $\Pr[i+1|j] = y\alpha^{j-i-1}$ and

$$\Pr[i|j] = y\alpha^{\lceil \frac{(j-i)+\min(j,n-j)}{2}\rceil} \geq y\alpha^{\lceil \frac{2(j-i)}{2}\rceil} = y\alpha^{j-i}$$

Similarly, we can show $\Pr[i|j] < y\alpha^{j-i}$. Hence we have

$$\alpha \Pr[i|j] \leq \Pr[i+1|j] \leq \Pr[i|j].$$

Case (4): $|i-j| < \min(j, n-j)$ but $|i+1-j| \geq \min(j, n-j)$.

Then we have $j < i$ and

$$i - j < \min(j, n - j) \leq i - j + 1$$

We have $\Pr[i|j] = y\alpha^{i-j}$ and

$$\Pr[i+1|j] = y\alpha^{\lceil \frac{1}{2}((i+1-j)+\min(j,n-j))\rceil} \geq y\alpha^{\lceil \frac{2(i+1-j)}{2}\rceil} = y\alpha^{i+1-j}$$

Similarly, we can show $\Pr[i|j] \leq y\alpha^{i-j}$. Hence $\alpha \Pr[i|j] \leq \Pr[i+1|j] \leq \Pr[i|j]$

Summary of Column Properties. When $i < j$, the column-wise adjacent probabilities are either the same or increase by a factor of $1/\alpha$ as i increases; and when $i \geq j$, then adjacent probabilities either decrease by a factor of $1/\alpha$ or stay the same. From these, we can conclude that EM has column monotonicity (and hence is column honest).

Row properties. The analysis for the row properties (DP, and row monotone) follows the pattern set by the column properties, based on a case analysis. Consider a fixed row i of the mechanism. As we look at neighboring entries j and $j + 1$ we have four cases:

Case (1): $|i - j| < \min(j, n - j)$ and $|i - (j + 1)| < \min(j, n - j)$.
Then $\Pr[i|j] = y\alpha^{|i-j|}$ and $\Pr[i|j + 1] = y\alpha^{|i-(j+1)|}$, so the probability either increases by a factor of $1/\alpha$ (when $j < i$) or decreases by a factor of $1/\alpha$ (when $j \geq i$).

Case (2): $|i - j| \geq \min(j, n - j)$ and
$|i - (j + 1)| \geq \min(j + 1, n - j + 1)$.
Then

$$\Pr[i|j] = y\alpha^{\lceil \frac{|i-j|+\min(j,n-j)}{2}\rceil}$$

while

$$\Pr[i|j + 1] = y\alpha^{\lceil \frac{|i-(j+1)|+\min(j+1,n-(j+1))}{2}\rceil}.$$

The subcases here are
(a) when $j \leq n/2$ and $j < i$. Then

$$\Pr[i|j] = y\alpha^{\lceil \frac{i-j+j}{2}\rceil} = y\alpha^{\lceil i/2\rceil} = \Pr[i|j + 1] = y\alpha^{\lceil \frac{i-j-1+j+1}{2}\rceil},$$

i.e. the probability is unchanged.
(b) when $j > n/2$ and $j > i$, then similarly

$$\begin{aligned}\Pr[i|j] &= y\alpha^{\lceil \frac{1}{2}(j-i+n-j)\rceil} \\ &= y\alpha^{\lceil \frac{1}{2}(j+1-i+n-j-1)\rceil} \\ &= \Pr[i|j + 1]\end{aligned}$$

Note that other potential cases, e.g. $j < i$ and $j \geq n/2$ are ruled out by the condition

$|i-j| \geq \min(j, n-j)$.

Case (3): $|i-j| < \min(j, n-j)$ but $|i-(j+1)| \geq \min(j+1, n-j-1)$. Working through the subcases eliminates most options: if $j < i$ we can derive $2(j+1) \leq i \leq 2j$, a contradiction. This leaves $j \geq i$, which leads us to

$$\begin{aligned} j - i &< n - j \\ j + 1 - i &\geq n - j - 1 \end{aligned}$$

Note that it must be that $|i-(j+1)| \geq n-j$, as the other possibility leads to $i > 2(j+1)$, contradicting $j \geq i$. Combining these two, we obtain $j < \frac{n+i}{2} \leq j+1$. Subtracting i from both sides, and applying the $\lceil \cdot \rceil$ operator, we obtain

$$\lceil j - i \rceil \leq \lceil \frac{n-i}{2} \rceil \leq \lceil j - i + 1 \rceil$$

Since j and i are both integral, we conclude

$$j - i \leq \lceil \frac{n-i}{2} \rceil \leq (j-i) + 1 \tag{6.16}$$

Then we have $\Pr[i|j] = y\alpha^{j-i}$, while

$$\Pr[i|j+1] = y\alpha^{\lceil \frac{1}{2}((j+1-i)+n-(j+1)) \rceil} = y\alpha^{\lceil \frac{n-i}{2} \rceil}.$$

From (6.16), we conclude that in this case

$$\alpha \Pr[i|j] \leq \Pr[i|j+1] \leq \Pr[i|j].$$

Case (4): $|i-j| \geq \min(j, n-j)$ but
$|i-(j+1)| < \min(j+1, n-j+1)$.
This case starts similarly to the previous case. We cannot have $i < j$ as this leads to a contradiction, so we must have $i \geq j$, and $j < n-j$. Then we deduce

$$\begin{aligned} i - j &\geq j \\ i - j - 1 &< j + 1 \end{aligned}$$

These permit only two possibilities: $i = 2j$ or $i = 2j+1$. In the first of these, we obtain

$$\begin{aligned} \Pr[i|j+1] &= y\alpha^{2j-(j+1)} = y\alpha^{j-1} \\ \text{and } \Pr[i|j] &= y\alpha^{\lceil \frac{1}{2}(j+j) \rceil} = y\alpha^{j}. \end{aligned}$$

Else, we obtain

$$\Pr[i|j+1] = y\alpha^{2j+1-(j+1)} = y\alpha^{j}$$
$$\text{and } \Pr[i|j] = y\alpha^{\lceil \frac{1}{2}(j+1+j)\rceil} = y\alpha^{j+1}$$

In both cases, we have $\Pr[i|j] = \alpha \Pr[i|j+1]$.

Summary of Row Properties. From the cases analyzed above, we see that when $i < j$, then adjacent probabilities are either the same or there is an increase by a factor of $1/\alpha$ as j increases; and when $i > j$, then adjacent probability either decreases by a factor of $1/\alpha$ as j increases, or stays the same. From these, we can conclude that EM meets differential privacy, and is row monotone.

These collectively cover all defined properties (due to implications discussed in Section 6.4, e.g. row monotonicity implies row-wise honesty). □

6.6.3 Comparing mechanisms

In Section 6.4, we define 7 different properties, denoted as RH, RM, CH, CM, WH, F, and S. We can seek a mechanism that satisfies any subset of these, suggesting that there are 128 combinations to explore. However, we are able to dramatically reduce this design space with the following analysis based on the $\mathbb{L}_0$ score function.

First, we have shown by Theorem 13 that EM has the optimal $\mathbb{L}_0$ score of any fair mechanism and has all other possible properties "for free". Therefore, for any desired set of properties that include F, we can just use EM. Second, we have shown by Theorem 12 that GM achieves symmetry and row monotonicity (and hence row honesty) at a cost which is optimal for any mechanism (i.e. BASICDP). Hence for any subset of {S, RM, RH}, it suffices to use GM.

In our experiments (Section 6.7.1), we show that there are only two remaining behaviours: either we solve the LP for the WH property alone, or we solve the LP for WH and CM properties. Both solutions come with symmetry (S) and row properties RH, RM at no additional cost. However, as noted in Lemma 13, GM satisfies WH when $n \geq \frac{2\alpha}{1-\alpha}$, so in this case, we can use GM. Last, from observations in Section 6.4, we have that CM $\Rightarrow$ CH $\Rightarrow$ WH, so any demand that requires any of these properties (and not F) can be satisfied by WM also. But in the weak privacy case that $\alpha \leq \frac{1}{2}$, GM has these properties, and so subsumes WM.

To summarize this reasoning, in the case that $\alpha \leq \frac{1}{2}$, there are only two competitive mechanisms: EM if fairness is required, and GM for all other cases. When $\alpha > \frac{1}{2}$, things are a little more complicated, so we show a flowchart in Figure 6.4: from 128 possibilities, there are only four distinct approaches to consider (two explicit mechanisms, and two solutions to an LP with different constraints), and the choice is determined primarily

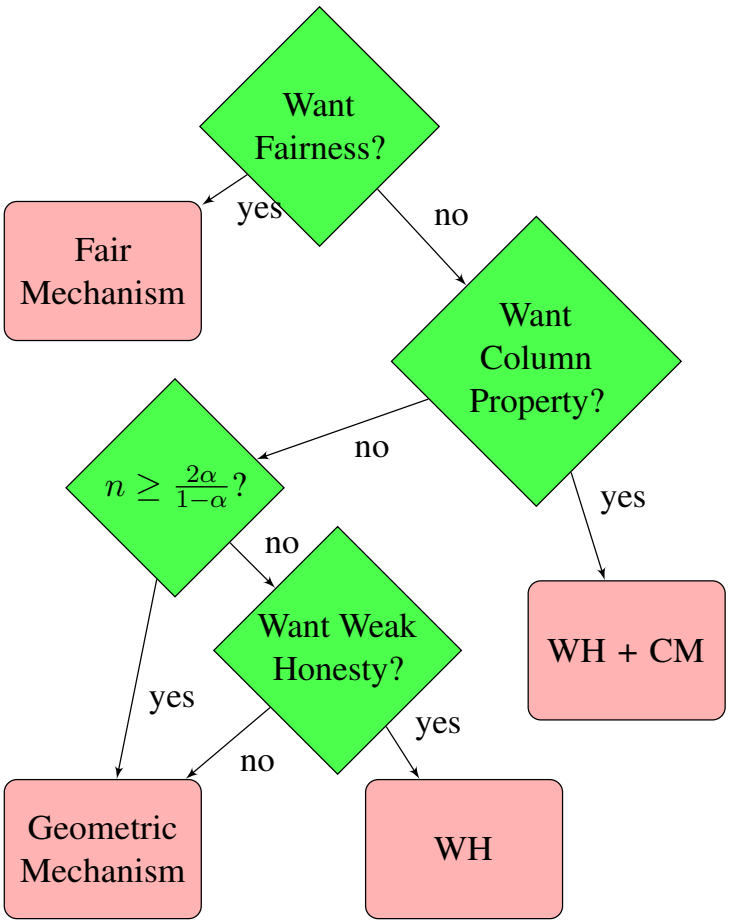


Figure 6.4: Flowchart of properties for $\mathbb{L}_0$ objective ($\alpha > \frac{1}{2}$).

by whether the mechanism is required to satisfy fairness, column properties, weak honesty, or none. We also consider the baseline method UM for comparison. We present a summary of these four named mechanisms in Figure 6.5: the explicit GM, UM and EM, and WM which is found by solving an LP. We write '—' for a property when this depends on the setting of the parameters (discussed in the relevant section). We see that EM has a very similar objective function value $\mathbb{L}_0$ (recalling that we are trying to minimize this value), and all the properties considered so far. We do not have a closed form for the $\mathbb{L}_0$ score of WM, as it is found by solving the LP; however it is no less than that for GM (since GM satisfies a subset of the required properties of WM), and no more than that of EM (since EM satisfies all properties).

At this point, we might ask how different are these mechanisms in practice — perhaps they are all rather similar? Figure 6.6 shows this is not the case for a small group size ($n = 4$). For a moderate value of the privacy parameter $\alpha = 0.9$, it presents the three non-trivial mechanisms using a heatmap to highlight where the large entries are. We immediately see that EM concentrates probability mass along a uniform diagonal (as

Property	GM	WM	EM	UM
Symmetry (S)	Y	Y	Y	Y
Row Monotone (RM)	Y	Y	Y	Y
Column Monotone (CM)	—	—	Y	Y
Fairness (F)	N	N	Y	Y
Weak Honesty (WH)	—	Y	Y	Y
$\mathbb{L}_0$	$\frac{2\alpha}{1+\alpha}$	$\geq \frac{2\alpha}{1+\alpha}$	$\approx \frac{2\alpha}{1+\alpha} \cdot \frac{n+1}{n}$	1

Figure 6.5: Properties of named mechanisms.

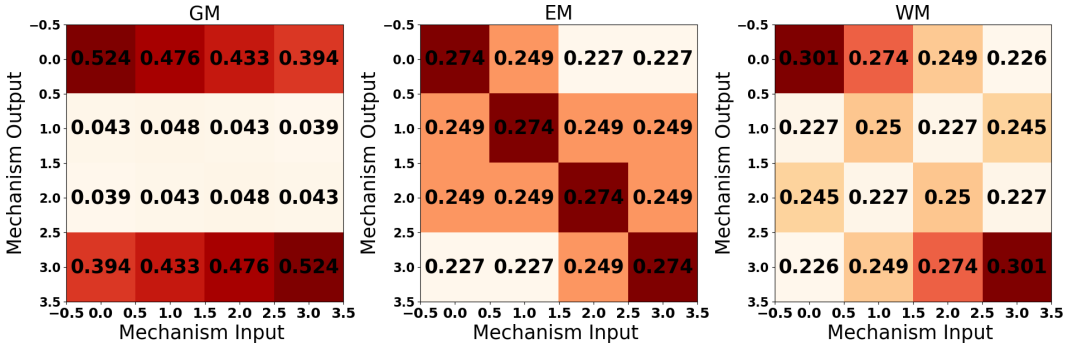


Figure 6.6: Heatmaps for GM, EM, WM with $n = 4$

required by fairness). Both GM and WM tend to favour extreme outputs (0 or 4 in this example) whatever the input, although GM is very skewed in this regard while WM is more uniform in allowing non-extreme outputs.

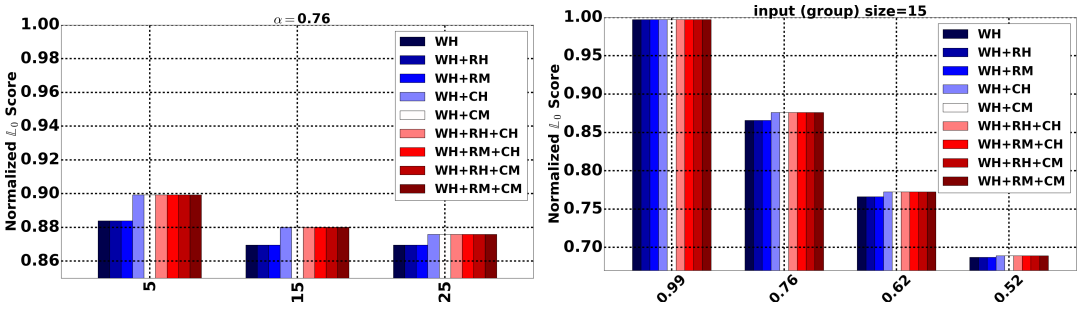
Last, we check that what we are doing is not a trivial modification of known mechanisms. Prior work [36, 37] showed how optimal unconstrained mechanisms can be derived from GM by transformations. Gupte and Sundarajrajan give a simple test: a mechanism $\mathcal{P}$ can be derived from GM iff every set of three adjacent entries in the mechanism satisfy

$$(\Pr[i|j] - \alpha \Pr[i|j-1]) \geq \alpha(\Pr[i|j+1] - \alpha \Pr[i|j])$$

We applied this test to mechanisms WM and verified that this condition is indeed violated for $n > 1$. For EM, this condition is automatically broken for all $n > 1$: we have $\Pr[2|0] = \Pr[2|1] = y\alpha$, while $\Pr[2|2] = y$. Then the condition is

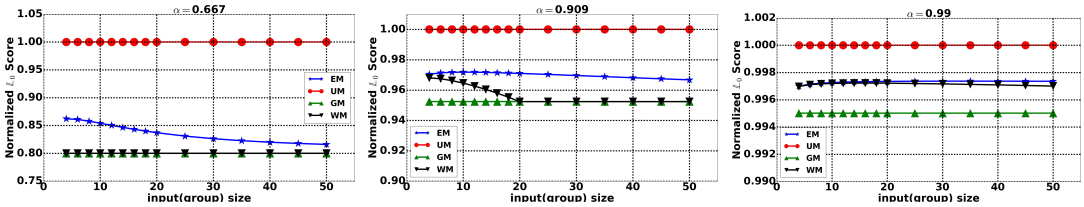
$$y\alpha(1-\alpha) \geq y\alpha(1-\alpha^2) \equiv 1 \geq (1+\alpha)$$

which is always false for $\alpha > 0$. Hence, these mechanisms are not derivable from GM.



(a) Varying group size (b) Varying α

Figure 6.7: Combinations of properties with Weak Honesty.



(a) $\alpha = \frac{2}{3}$ (b) $\alpha = \frac{10}{11}$ (c) $\alpha = \frac{99}{100}$

Figure 6.8: Final groups of mechanisms with distinct behaviours.

6.7 Experimental Evaluation

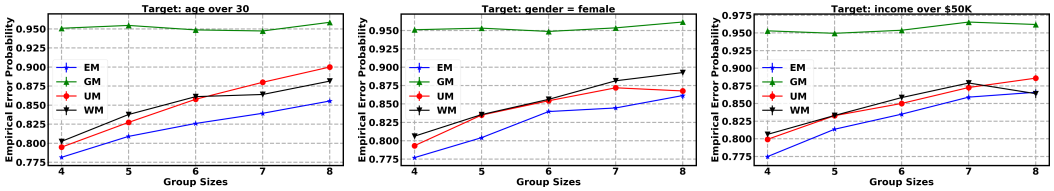
The purpose of our experimental study is two-fold. In Section 6.7.1, we substantiate our earlier claims about properties of mechanisms satisfying weak honesty (but not fairness). In what follows, we look at other measures of utility of these mechanisms, to understand their robustness.

Default Experimental Settings. All experiments in this work were implemented in Python, making use of the standard library NumPy to handle the linear algebraic calculations, and PyLPSolve [141] to solve the generated LPs. Evaluation was made on a commodity machine running Linux. We omit detailed timing measurements, as the time to solve the LPs generated was negligible (sub-second).

Experimental Setting. We considered a variety of settings of parameter α (typical values chosen are $\{\frac{1}{2}, \frac{2}{3}, \frac{10}{11}, \frac{99}{100}\}$ and group size n (ranging from 2 up to hundreds).

6.7.1 $\mathbb{L}_0$ Objective Function

Our first experiment analyzes the effect of weak honesty combined with other properties drawn from {CH, CM, RH, RM}, including the empty set. There are 9 meaningful combinations of properties to ask for, which we write as {$\emptyset$, RH, RM, CH, CM, RH+CH, RH+CM,



(a) Estimating young population (b) Estimating gender balance (c) Estimating income level

Figure 6.9: Empirical Error Probability on Adult Dataset for $\alpha = 0.9$.

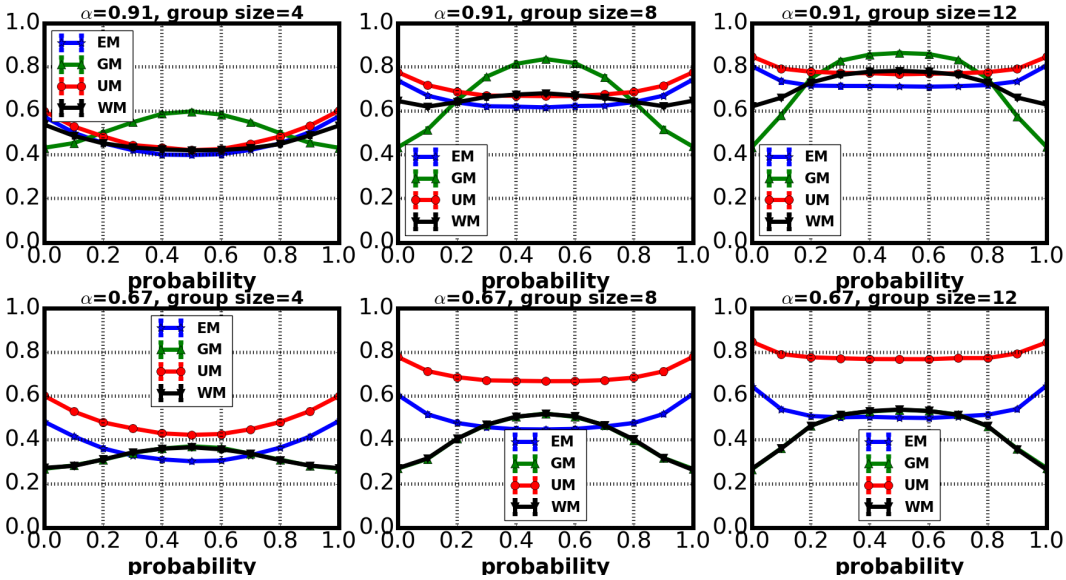


Figure 6.10: $\mathbb{L}_{0,1}$ score for Binomial data, for $n = \{4, 8, 12\}$ and $\alpha = \{0.91, 0.67\}$.

RM+CH, RM+CM} — other combinations reduce to these, since RM implies RH, and CM implies CH.

As discussed in Section 6.6.3, there are cases when the solution found by solving the LP has cost $\frac{2\alpha}{1+\alpha}$ and is identical to GM: these are when $n \geq \frac{2\alpha}{1-\alpha}$ and only row-wise properties are requested, consistent with Figure 6.4. This is borne out in Figure 6.7: we see that when WH alone is requested, or in combination with only row properties (RH or RM) we get a lower $\mathbb{L}_0$ value than when any column properties (CH or CM) are requested. Figure 6.7(a) shows the case for different values of n. When $n > \frac{2\alpha}{1-\alpha}$, which is 6.33 in this example (where $\alpha = 0.76$), the cost of WH alone is $\frac{2\alpha}{1+\alpha} = 0.864$, the cost of GM. For large α (Figure 6.7(b)), the cost of all combinations of WH are the same, and identical to the cost of EM; as α is decreased, we see two behaviours, where the lower $\mathbb{L}_0$ cost is that of GM. We confirmed this behaviour for a wide range of n and α values. From now on, we use WM to refer to the mechanism with WH, RM and CM properties.

The relationship between the $\mathbb{L}_0$ scores for the three mechanisms is further clarified in Figure 6.8. The plots show the $\mathbb{L}_0$ scores of GM, WM, EM and UM for different

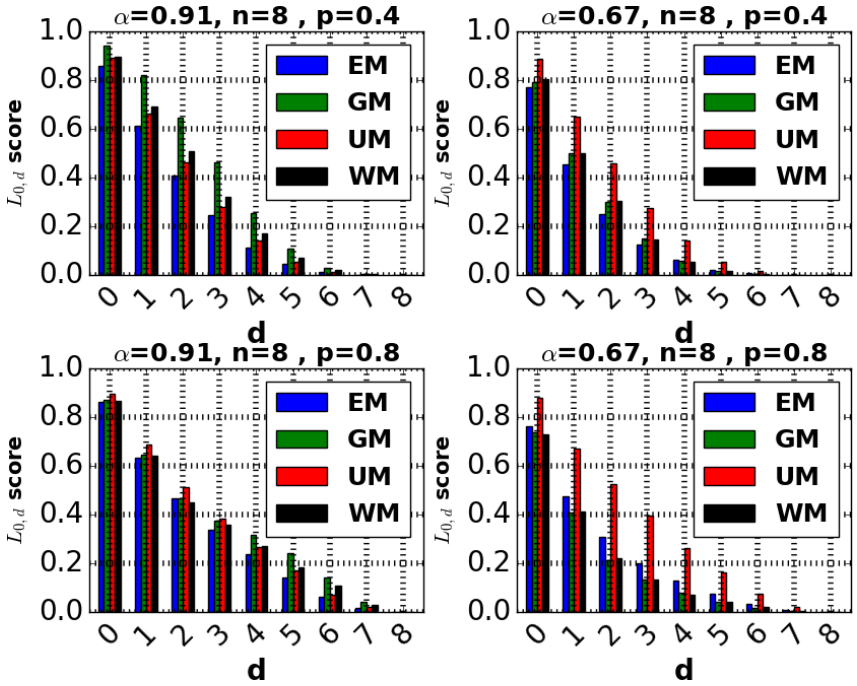


Figure 6.11: Histograms of $\mathbb{L}_{0,d}$ scores for binomial data.

values of α. In Figure 6.8(a), $\alpha = \frac{2}{3}$ so the threshold $\frac{2\alpha}{1-\alpha} = 4$. Then GM satisfies WH for the whole range of n values shown, so WM converges on GM, while EM has a higher (but decreasing) cost. For Figure 6.8(b), $\alpha = 10/11$ so the threshold is 20. Indeed, we see that the cost of WM converges with GM at $n = 20$. Last, in Figure 6.8(c), the threshold of 198 is far above the range of n values shown, so WM does not converge on GM here. Rather, for this high value of α, the y value for EM is above $\frac{1}{n+1}$ for all n: so in this case EM has weak honesty, and the cost of WM remains the same as that of the optimal fair EM.

Now we verify the performance of our mechanisms on real and synthetic datasets. In these experiments, the users are partitioned into small groups and the goal is to release the group totals privately under ϵ-DP/$(\epsilon, 1)-$LLDP. Unlike LDP, our goal is not to aggregate the histograms but only to verify the deviation of the perturbed values from the truth by various error metrics.

6.7.2 Experiments On Real Data

We make use of the UCI Adult dataset, a workhorse for privacy experiments [142]. Our instance of the dataset contains demographic information on 32K adults with 15 columns listing age, job type, education, relationship status, gender, and (binary) income level. We created three binary targets, treated as sensitive: income level (high/low), gender (male/female), and young (age over/under 30). To form small groups, we gathered the rows (corresponding to individuals) arbitrarily into groups of a desired size.

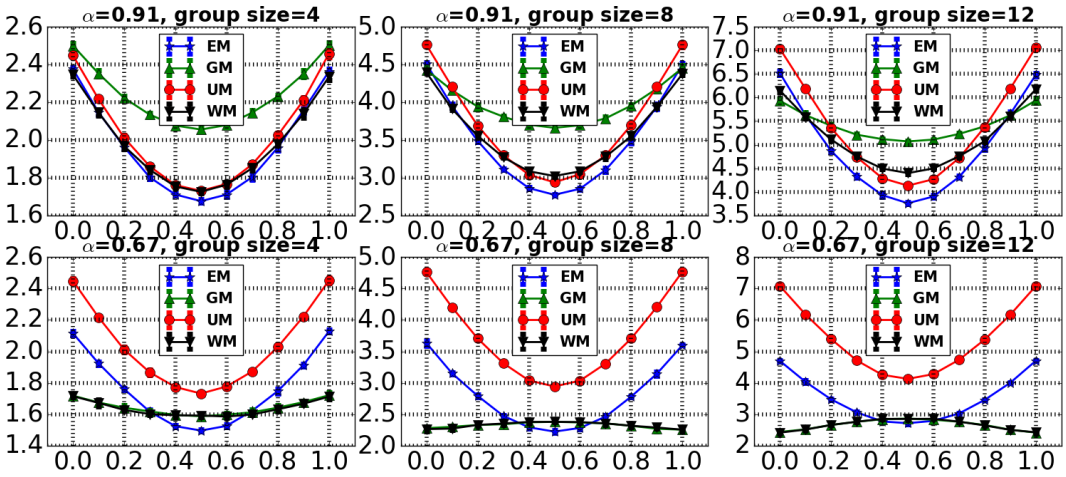


Figure 6.12: Root mean square error plots for binomial data.

Figure 6.9 shows results for the $\mathbb{L}_0$ objective, that is, where we focus on the fraction of times the mechanism reports an incorrect answer, as a function of group size. Specifically, we count the number of groups whose noisy count for each target attribute is not equal to their true count. We expect this quantity to be fairly high, as it measures how often our mechanism is honest, i.e. returns the true input. Other experiments (not shown) computed the corresponding probability for returning an answer that is close to the true one, e.g. off by at most one, and showed similar patterns. The plot includes error bars from 50 repetitions of this process to show 1 standard error.

Observe first that the performance of UM is essentially independent of the input data: the chance of it picking the correct answer is always $1 - \frac{1}{n+1}$ for a group of size n, and indeed we see this behaviour (up to random variation). We would hope that our optimized mechanisms can outperform this trivial method. Perhaps surprisingly, on this data GM does appreciably worse. This highlights the limitations of GM. In this data, the common inputs are around the middle of the group size (i.e. typically close to $n/2$). It is on these inputs that GM does poorly, and only does well for inputs that are 0 or n, which happen to be rare in this dataset (in other words, the data distribution does not match the prior for which GM is optimal). The condition of weak honesty is not sufficient to improve significantly over random guessing: for this data, we see that WM tracks UM quite closely. It is only the most constrained mechanism that fares better on this evaluation metric for this data: EM which achieves fairness gives the best probability of returning the unperturbed input. In corresponding experiments with higher values of α in the range 0.9 to 0.99, corresponding to the strongest privacy guarantees adopted in prior work on differential privacy, there is not much to choose between EM and WM, and it gets even harder to show substantial improvement over uniform guessing. In order to understand the behaviours of the mechanisms further, we next consider synthetic data, where we can directly control the

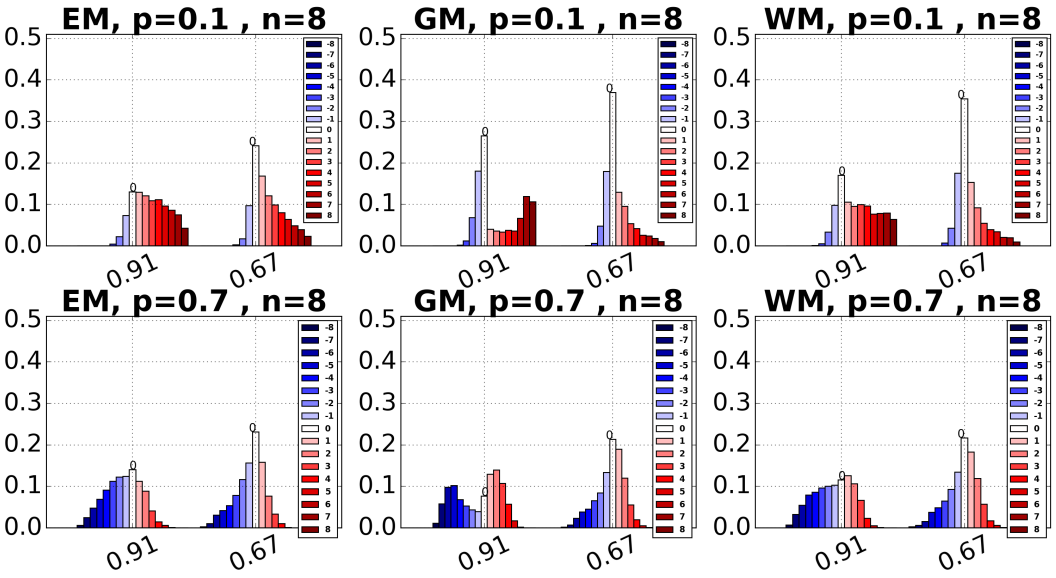


Figure 6.13: Error histograms on group size 8 for $p = 0.1$ and $p = 0.7$, with $\alpha = \{0.91, 0.67\}$.

data skewness within groups.

6.7.3 Experiments On Synthetic Data

In our experiments with synthetic data we generate a population of $10,000$ individuals each with a private bit and divide them into small groups of the same size, n. Each individual has the same probability p of having their bit be one, so the distribution within each group is Binomial. Hence, the expected count for each group is pn. Our experiments vary the parameters p, n and α.

$\mathbb{L}_{0,1}$ **Error.** Our experiments so far have used the target objective function $\mathbb{L}_0$ to evaluate the quality of the mechanism. This is sufficient to distinguish the different mechanisms, but all mechanisms achieve a score which is still quite close to 1, obtained by uniform guessing. To better demonstrate the usefulness of the obtained mechanisms, we use other functions to evaluate their accuracy. Figure 6.10 uses the related measure of $\mathbb{L}_{0,1}$ i.e. the fraction of groups which output a value differing from their true answer by more than 1, as we vary data distribution (determined by p), group size n, and privacy parameter α. We stress that though we use $\mathbb{L}_{0,1}$ for evaluation, we continue to use mechanisms designed for minimizing the $\mathbb{L}_0$ error. Each subplot in the figure represents a configuration of $\langle \alpha, n \rangle$, describing how $\mathbb{L}_{0,1}$ error changes with input distribution parameter p. Each experiment is repeated 30 times and we observe that the results have very small variance.

It is apparent that the shape of the input distribution has a pronounced effect on the quality of the output. We confirm that GM can do well when the input is very biased

(p close to 0 or 1), which generates more instances with extreme input values. However, when the input is more spread across the input space, the more constrained mechanisms consistently give better results. For higher α, the constrained methods have similar behaviour, and improve only slightly over UM (while GM is often worse than uniform). Enforcing fairness tends to make EM less sensitive to the input distribution, except when the input is an extreme value (0 or n). When α is lower (second row), the overall scale of error decreases and WM and GM converge, as noted previously.

$\mathbb{L}_{0,d}$ Error. In the previous experiment, we fixed $d = 1$ and evaluated our mechanisms for variety of input distributions. Next we vary d while holding input size and input distributions steady, and compute $\mathbb{L}_{0,d}$ error. Figure 6.11 plots the fraction of population reporting a value that is more than d steps away from the true answer for various d values with $n = 8$. This captures the probability mass in the tail of each mechanism.

In the top row, we use a more proportionate input distribution. Here, EM outperforms all other mechanisms, sometimes by a substantial fraction. Interestingly, the margin between EM and GM only increases with larger d. Once again we see that for higher α values, use of GM can yield accuracy worse than mere random guessing. For lower α's GM's accuracy increases dramatically but still remains worse than EM's.

In the bottom row, the input distribution is more skewed, which tends to favour GM. However, EM does not do substantially worse than GM even for this biased input distribution. The intermediate mechanism found by WM tends to fall between GM and EM. We observed similar behaviour for other values of n.

Root Mean Square Error (RMSE). Our next set of experiments compute the RMSE error (a measure of variance and bias of a mechanism) of estimates from small groups. Note that none of our mechanisms are designed to optimize this metric, but we can nevertheless use it as a measure of the overall spread of error. Figure 6.12 shows plots with error bars showing one standard deviation from 30 repetitions.

As seen in previous experiments, a more symmetric input distribution (p closer to 0.5) tends to be easier for most mechanisms — although we see cases where GM finds this more difficult. Increasing the group size increases the RMSE, as there is a wider range of possible outputs, and the constraints ensure that there is some probability of producing each possible. Yet again, we see that increasing α tends to make GM less competitive and find many cases where GM is worse than random guessing (UM). The interesting case may be for fairly high privacy requirements ($\alpha = 0.91$), where we observe that EM tends to give lower error across all group sizes and input distributions.

Error Histograms. As previously discussed, the measure of error probability gives some insight into the difference between mechanisms, but holds them to a high standard. This probability is high for all mechanisms, as we do not expect them to give the exact correct

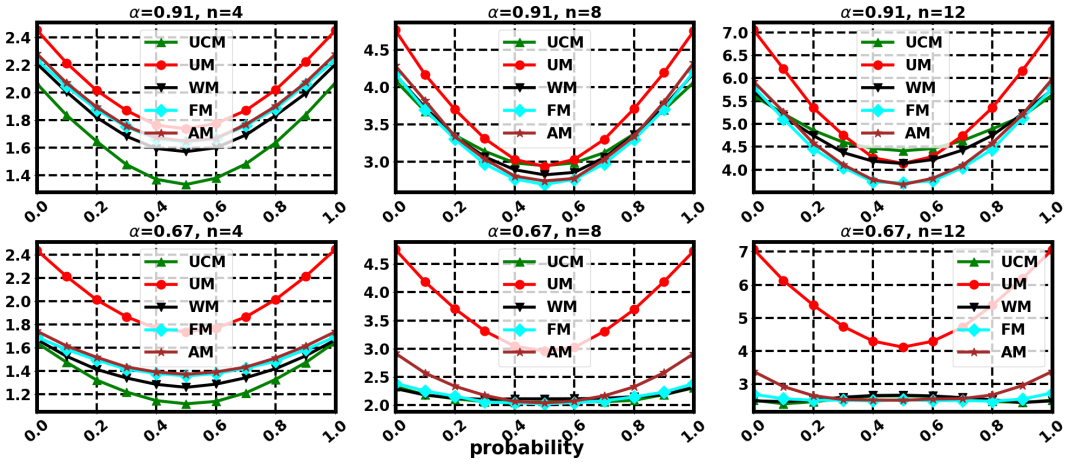


Figure 6.14: Root mean square error plots on Binomial data for $\mathbb{L}_1$ objective function mechanisms.

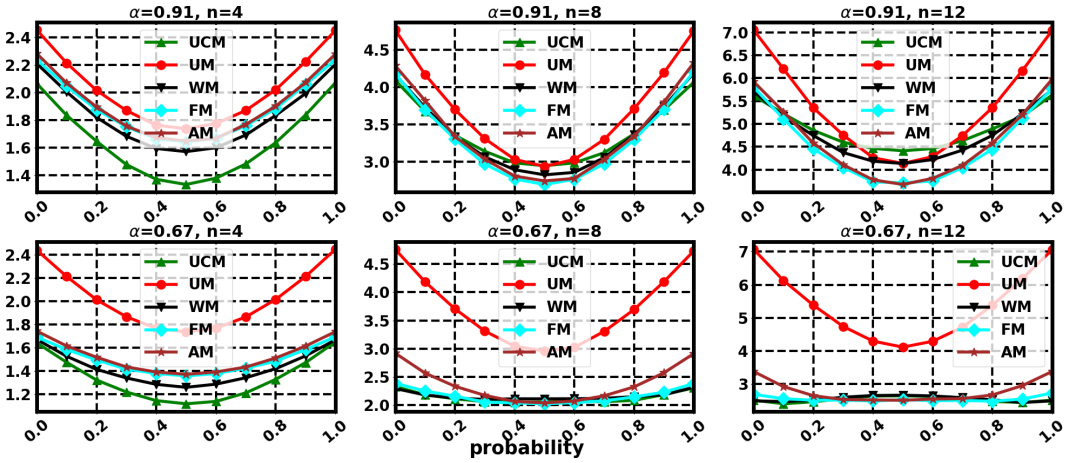


Figure 6.15: Root mean square error plots for binomial data for $\mathbb{L}_2$ objective function mechanisms.

answer. To see the spread of error from another perspective, we plot error histograms for our mechanisms — for a given input distribution, how often is the response correct, how often is it an overestimate by one, and so on. For example, when an input of 1 is reported as 0, the error is -1. Figure 6.13 shows the error histograms for a representative group size $\{8\}$ with $p = 0.1$ and $p = 0.7$ for two extreme α values. For each case, we show error histograms for the three mechanisms EM, GM, and WM.

In the $p = 0.1$[2] case, the input does not permit significant underestimation (most true answers are small). All mechanisms are more likely to give zero error. This is enforced by the fairness (for EM) or weak honesty (WM) properties. For GM, we see that for $\alpha = 0.91$, there is a second peak corresponding to an output of n. So it tends to have a larger

[2] p is a parameter for producing synthetic input data introduced in 6.7.3.

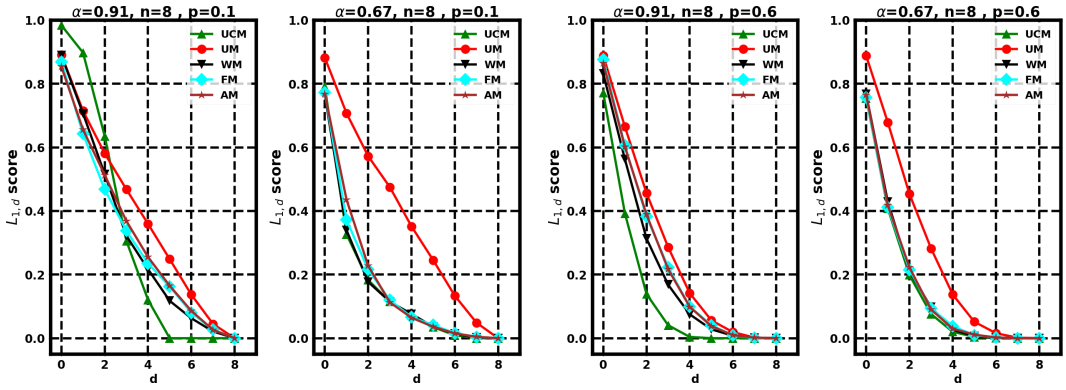


Figure 6.16: Line plots for $\mathbb{L}_{1,d}$ scores for binomial data ($p = 0.1$ and $p = 0.6$).

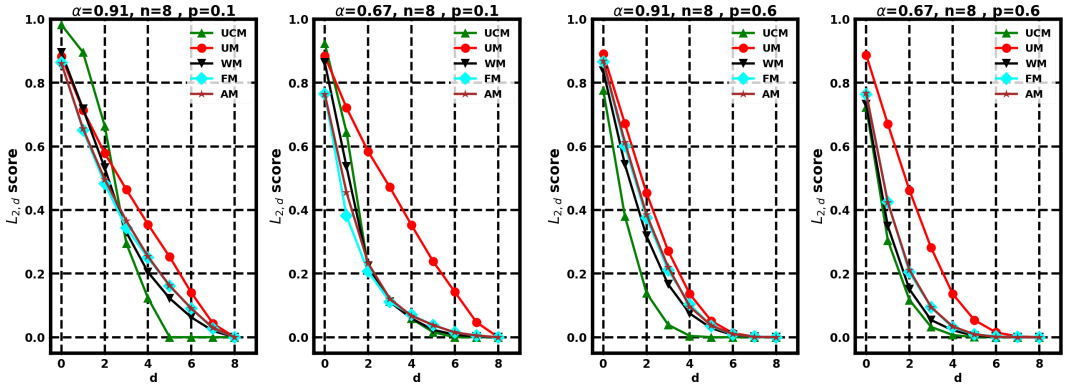


Figure 6.17: Line plots for $\mathbb{L}_{2,d}$ scores for binomial data ($p = 0.1$ and $p = 0.6$).

error when it does not output the true answer. We observe that the column monotonicity properties of EM and WM tend to force a smoother error distribution.

Some similar behaviour is observed for $p = 0.7$. Here, the support of the input distribution is broader, and hence so is the support of the error distribution. We still observe that GM tends to have a bimodal error distribution for high α, with a dip around zero error. As α decreases, the mechanisms become more similar, in particular WM tends to look more like GM (as we have seen, they converge to the same mechanism for $n \geq \frac{2\alpha}{1-\alpha}$). We have observed similar trends for other n values.

6.7.4 $\mathbb{L}_1$ and $\mathbb{L}_2$ objective functions.

We have already seen that unconstrained mechanisms for $\mathbb{L}_1$ and $\mathbb{L}_2$ can have pathological outcomes. In this section, we return to these objective functions, and study their behaviour under the imposition of conditions. In contrast to the $\mathbb{L}_0$ case, we observed that the number of distinct mechanisms obtained under selection of different subsets of conditions was quite

large. In order to constrain the number of mechanisms under consideration, we restrict our attention to a small number of options: enforcing Weak Honesty (denoted WM) or Fairness (denoted FM) only; or requiring either no properties at all (the unconstrained mechanism, UCM), or all properties simultaneously (the all properties mechanism, AM). We also compare to the trivial uniform mechanism (UM) for calibration. Among these four options, we expect UCM to obtain the lowest error since it can directly optimize the target function, with the comensurate disadvantages discussed previously.

Root Mean Square Error (RMSE). Figures 6.14 and 6.15 show plots for the root mean square error on binomially distributed input data, similar to Figure 6.12. For this measure of accuracy, UCM provides among the best results. However, for small groups, we would tend to prefer WM, since it provides a similar level of accuracy while avoiding the degerate behaviours. AM performs well when the input distribution is close to the symmetric case ($p = 0.5$), but has weaker results when the input is more skewed (smaller or larger p values). FM is observed to do better as the group size increases.

$\mathbb{L}_{1,d}$ and $\mathbb{L}_{2,d}$ functions. Figures 6.16 and 6.17 show plots of the errors for the functions $\mathbb{L}_{1,d}$ and $\mathbb{L}_{2,d}$ respectively, similar to Figure 6.11 for $\mathbb{L}_{0,d}$. When $p = 0.1$, most groups have sums close to 0. For larger α's and smaller d's, UCM performs worst. This situation is reversed as d increases. That is, the mechanism reduces the probability mass that is far from the true answer, and the expense of increasing the mass close to the true answer but not equal to it. It tends to map most inputs to outputs close to $\lfloor \frac{n}{2} \rfloor$. UCM is then the preferred mechanism for more balanced distributions ($p = 0.6$). AM, FM and WM all behave quite similarly to each other, and their lines almost overlap for smaller α. In summary, AM, FM are slightly preferable for skewed input distributions and strong privacy requirements, whereas WM is suitable in general for distributions with less bias.

6.8 Linear Programming Framework For LDP

It is possible to extend Ghosh *et al.*'s LP framework to LDP with a simple modification and the returned mechanism can be used for histogram aggregation.

Model. We consider the local model once again. Each user has an item from $\{0, .., n-1\}$. [3] Our goal is to design a LDP compliant $n \times n$ matrix that can be used as a mechanism to estimate the frequency of each item $j \in [n]$. Equation 6.17 to 6.20 denote the LP framework

[3] We represent the domain size by n instead of D to remain consist with the notations used in this chapter.

to design the LDP mechanisms for histogram aggregation.

$$\text{minimize:} \quad \sum_{j=0}^{n} w_j \sum_{i=0}^{n} |i-j|^p \rho_{i,j} \tag{6.17}$$

$$\text{subject to:} \quad 0 \leq \rho_{i,j} \leq 1 \quad \forall i,j \in [n] \tag{6.18}$$

$$\sum_{i=0}^{n} \rho_{i,j} = 1 \quad \forall j \in [n] \tag{6.19}$$

$$\rho_{i,j} \geq \alpha \rho_{i,k}, \text{ and } \rho_{i,k} \geq \alpha \rho_{i,j} \quad \forall i,j,k \in [n] : j \neq k \tag{6.20}$$

The first three equations remain the same in LDP. Note the difference in equation 6.20 due to LDP constraints. Since any pair of numbers $j, k \in [n]$ are in the neighborhood, the indices in the conditional probabilities change. For $p = 0$, the LP solver returns the following interesting mechanism.

$$\begin{pmatrix} \mathbf{p} & \frac{1-p}{n-1} & \frac{1-p}{n-1} & \frac{1-p}{n-1} & \cdots & \frac{1-p}{n-1} \\ \frac{1-p}{n-1} & \mathbf{p} & \frac{1-p}{n-1} & \frac{1-p}{n-1} & \cdots & \frac{1-p}{n-1} \\ \frac{1-p}{n-1} & \frac{1-p}{n-1} & \mathbf{p} & \frac{1-p}{n-1} & \cdots & \frac{1-p}{n-1} \\ \frac{1-p}{n-1} & \frac{1-p}{n-1} & \frac{1-p}{n-1} & \mathbf{p} & \cdots & \frac{1-p}{n-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1-p}{n-1} & \frac{1-p}{n-1} & \frac{1-p}{n-1} & \frac{1-p}{n-1} & \cdots & \mathbf{p} \end{pmatrix}$$

Here $p = \frac{1}{1+(n-1)\alpha}$. Observe that this mechanism is equivalent to GRR. GRR already satsifies all properties we have proposed without explicit enforcement. We observed that forcing these properties did not change shape of the mechanism.

Remarks About GRR. An immediate question to think about is – despite its optimality, why is GRR significantly inaccurate compared to other mechanisms such as OUE, OLH and HRR? The answer lies in the formulation of the linear program. The above formulation only optimizes for a given loss function in the space of mechanisms with the same domain and the range size i.e. n[4]. And GRR is indeed the most accurate mechanism under this constraint. However, none of the other mechanisms have this constraint. For example, OUE maps an item $i \in [n]$ (originally represented with $\log n$ bits) to a n bit vector. Similarly, HRR summarizes an n bit item with a single bit.

Constrained Optimization Under LDP. Optimizing for $\mathbb{L}_1$ and $\mathbb{L}_2$ under LDP constraints on the other hand produced pathological mechanisms similar to the ones in figure 6.1. It is possible to remove these anomalies by forcing a subset of properties. However, this is not an

[4]This is why we can represent GM/GRR as a square matrix.

immediately attractive avenue to pursue since the obtained mechanisms are only optimal for a given loss function when the range and the domain are the same. We can always design better mechanisms in local setting without this constraint as we have seen. It may be possible to modify this framework to produce *rectangular* mechanisms but thorough understanding of this direction requires a separate investigation.

Chapter 7

Conclusions and Impact Statement

7.1 Summary and Future Work

7.1.1 Marginal Queries

In chapter 4, we have provided algorithms and results for the central problem of private release of marginal statistics on populations. Our main conclusion is that methods based on Fourier (Hadamard) transformations of the input are effective for this task, and have strong theoretical guarantees in terms of accuracy, communication cost, and speed. Although the technical analysis is somewhat involved, the algorithms are quite simple to implement and so would be suitable for inclusion in current LDP deployments. Here is a direction worth pursuing in future.

Orthogonal Decomposition. It is natural to ask whether there are alternative decompositions for categorical data which share many of the properties of the Hadamard transform (orthogonal, requiring few coefficients to reconstruct low-order marginals). One such approach is the *Efron-Stein decomposition* [143] which is a generalization of Hadamard transform for non binary contingency tables. Similar to HT, it is possible to extract a set of Efron-Stein coefficients necessary and sufficient to evaluate a full set of a $k-$way marginals. One could then design an algorithm similar to INPHT that adds noise to a random coefficient, allowing an unbiased estimate to be constructed by an aggregator. We conjecture that for low order marginals, a scheme based on such decomposition will be among the best solutions.

7.1.2 Range Queries

In chapter 5, we have seen that we can accurately answer range queries under the model of local differential privacy. Two methods whose counterparts have quite differing behaviour in the centralized setting are very similar under the local setting, in line with our theoretical analysis. We sketch four possible extensions for future work:

Multidimensional range queries. Both the hierarchical and wavelet approaches can be extended to multiple dimensions. Consider applying the hierarchical decomposition to two-dimensional data, drawn from the domain $[D]^2$. Now any (rectangular) range can be decomposed into $4(B-1)^2 \log_B^2 D$ B-adic rectangles (where each side is drawn from a B-adic decomposition), and so we can bound the variance in terms of $(B-1)^4 \log_B^4 D$. More generally, we achieve variance depending on $((B-1)\log D)^{2d}$ for d-dimensional data. Similar bounds apply for generalizations of wavelets. These give reasonable bounds for small values of d (say, 2 or 3). For higher dimensions, we anticipate that coarser gridding approaches would be preferred, in line with [121].

Advanced data analysis. Many tasks in data modeling and prediction can abstractly be understood as building a description of observed data density. The statistic of area under curve (AUC) defined in Section 2.5.3 is one such example. Recall that the AUC statistic is the empirical probability that a randomly sampled positively labeled score z_i is larger than a randomly sampled negatively labeled score z_j.

Consider an environment with each user i holding a pair $(z_i, y_i) \in [-d, d] \times \{-1, 1\}, d < \infty$. We discretize the scores from $[-d, d]$ to $[D]$ and aggregate the score histograms $H^+, H^- \in \mathbb{R}^D$ corresponding to the two labels. Assuming that the bins in the histograms are arranged in sorted order, the equation 2.12 can be rewritten as below.

$$AUC = \frac{\sum_{i=0}^{[D]} \sum_{j=0}^{i-1} H^+[i] \times H^-[j]}{\sum_{i=0}^{[D]} H^+[i] \sum_{j=0}^{[D]} H^-[j]} \tag{7.1}$$

In equation 7.1, we are simply multiplying the value of CDF $[0, i-1]$ in H^- to the frequency of ith score in H^+. Consider the local environment once again. Our goal is to estimate the AUC in a non-interactive fashion under a LDP guaratnee. Assuming that the labels are private too, a naive way could be to aggregate the flat joint histogram of size $2D$ using a FO. The error in aggregation of AUC due to the noise is $\mathcal{O}(D^2 \mathcal{V}_F)$. Alternatively, using HH or HaarHRR to find the required CDF's by answering the prefix queries would reduce the error to $\mathcal{O}((B-1)D\mathcal{V}_F \log_B^2 D)$. $\mathcal{V}_F$ here is the variance incurred in the estimation of the multiplication of two bin counts. The discretization error however would remain the same in both the approaches. Increasing D would reduce the discretization error but increase the error due to noise. Therefore, the first key challenge is to analytically find an expression for error in estimation as a function of D under suitable smoothness assumptions on the input distribution. Then one can find a D that balances the two errors.

Range Mean Estimation In Key-Value Pairs Setting. Our proposals can also be extended to the setting when the data are key-value pairs i.e. each user has a private integer valued key $z_i \in [D]$ and a corresponding numeric value $v_i \in [-1, 1]$. An interesting question to study is to find the mean of values of the keys in a given range $[a, b]$ under LDP constraints. More

formally,

$$M_{[a,b]} = \frac{\sum_{i=1}^{N} v_i \times \mathbb{I}_{a \le z_i \le b}}{\sum_{i=1}^{N} \mathbb{I}_{a \le z_i \le b}}$$

Note that the accuracy of mean estimation depends on the accuracy of range estimation. Here is our flat baseline solution. We first have each user discretize the values into $\{-1, 1\}$ through this rounding scheme — $\Pr[v'_i = -1|v_i] = \frac{1-v_i}{2}, \Pr[v'_i = 1|v_i] = \frac{1+v_i}{2}$. Then we estimate the joint histogram of size $2D$ using a FO. Let $\widehat{f}_j^{-1}$ and $\widehat{f}_j^{1}$ denote estimated frequency of -1's and 1's for a key $j \in [D]$. It is easy to see that an unbiased estimation for the sum of the values corresponding to the key j is given by $\widehat{f}_j^{1} - \widehat{f}_j^{-1}$. Therefore, the $\widehat{M}_{a,b}$ is estimated as below.

$$\widehat{M}_{[a,b]} = \frac{\sum_{a \le b} \widehat{f}_a^{1} - \widehat{f}_a^{-1}}{\sum_{a \le b} \widehat{f}_a^{1} + \widehat{f}_a^{-1}}$$

Once again, the variance of estimation in this flat approach increases linearly with the range size. Instead, we can slightly modify HH to maintain $2^{\ell+1}$ bins (instead of 2^{ℓ}) at each level $\ell \le \log_B D$ and evaluate the mean of a range of length r using atmost $4(B-1)\log_B(r)\log_B D$ nodes. We conjecture that this solution will be among the best performing ones. This problem becomes more challenging when users hold a set of key value pairs of unequal lengths and requires a fresh investigation.

Range Queries In Itemset Setting. Throughout chapter 5, we have assumed for simplicity that each user holds only a single item. In more realistic scenarios, users may have a set of items of unequal lengths. Sets of differing sizes pose a great challenge at aggregator's end in estimation. We now sketch a first-cut approach to extend our solution to deal with this requirement. Our goal in this case is to identify and evaluate only the heavy intervals.

Qin *et al.* [81] proposed an interative solution for finding top-k heavy hitters in the set valued setting i.e. each user has a private set of items. We adapt some of the elements of their total solution in our case. We can have each user pad their set s_i with dummy items to ensure that all sets are of the same preagreed cardinality ℓ_1. Typically ℓ_1 is set to a value larger than 90% of the set sizes. In the first round, each user perturbs a randomly sampled item from s_i with budget $\frac{\epsilon}{2}$ using a FO. Based on the frequencies of the items, the aggregator then prepares a set of potential heavy hitters items H and shares it with the users. In the next round, users find the heavy hitter items they have by performing the intersection $s_i \cap H$ and append this list with additional items from H to be make it of length $\ell_2 = 2k$. Finally, user invokes HH or HaarHRR for a randomly sampled heavy hitter item with $\frac{\epsilon}{2}$. The aggregator can set the count for the non heavy hitter items to 0 and ignore the queries that involve all non-zero counts. This is a good baseline but there is a lot of scope for improvement since

we are implicitely assuming that heavy intervals only consists of top-k elements which may not be the case. Besides, theoretical measurement of the quality this heuristic is far from immediate.

7.1.3 Count Queries

Chapter 6 proposed and studied several structural properties for privacy preserving mechanisms for count queries. We show how any combination of desired properties can be provided optimally under $\mathbb{L}_0$ by one of a few distinct mechanisms. Our experiments show that the "optimal" GM often displays the undesirable property of tending to output extreme values (0 or n). In practice, this means it is often not the mechanism of choice, particularly when α is large (above 0.7), but can be acceptable for smaller privacy parameters. EM and WM are quite different in structure, but are often similar in performance.

It is natural to consider other possible properties—for example, one could imagine taking a version of the DP constraint applied to columns of the mechanism (in addition to the rows): this would enforce that the ratio of probabilities between neighboring *outputs* is bounded, as well as that of neighboring inputs. The next logical direction is to provide a deeper study of mechanisms with various properties using $\mathbb{L}_1$ or $\mathbb{L}_2$ as objective function, building on our empirical observations. It will be interesting to study tailor-made linear programming mechanisms that aim to optimize other queries such as range queries.

7.2 Impact Statement

This thesis made some timely contributions to the large stream of works on differential privacy that appeared in the last five years. Here is how we think the advancements proposed in this thesis influence the overall landscape of DP.

- Through the case studies of Hadamard transform and discrete Haar transform, we emphasize the importance of reconsidering the approaches previously proposed in the centralized model. Both the transforms were originally proposed in the centralized case but superseded by more sophisticated approaches. Interestingly, these discarded transforms re-emerge and in fact become the most preferred methods in the local case.

- We establish and promote the Hadamard transform based solution as a generic histogram aggregation primitive in chapter 5. We are confident that this primitive will be used as the preferred method in the follow up works since it has the same accuracy as OUE and OLH without their drawbacks.

- It is expected that both Hadamard and discrete Haar transform will find more applications in the LDP contexts due to their rich mathematical properties.

- A line of research in DP attempts to seek the possibility of designing *generic all purpose* mechanisms optimizing on a loss function such as expected utility. The idea is that the data released via such mechanisms could be used for multiple potentially independent analysis tasks hoping that these generic mechanisms will be fit for most tasks. We build a strong case against such *one-size-fits-all* mechanisms by revealing the weaknesses in GM and the linear programming framework. Our study demonstrates the value of understanding the properties of the dataset (e.g. input distribution) before employing *of-the-shelf mechanisms* and developing solutions tailored for the task at hand.

Finally, we conclude this dissertation by spelling out the limitations of our study.

7.3 Limitations

- Our solutions developed for marginal and range queries require a large number of users (or much larger privacy budgets) to achieve acceptable values of accuracy. This means these methods are not suitable for smaller datasets. However, this in general is a well-known limitation of the local model.

- Throughout the dissertation, we assume that each data point is generated independently from the same distribution. Furthermore, each participant contributes exactly to a single record in the dataset and the dataset does not undergo any updates once gathered. While these assumptions are commonplace in a large body of LDP literature; it limits the applicability of these solutions including ours. However, we conjecture that a similar algorithmic building blocks will used in the solutions designed for the scenarios where these assumptions are relaxed.

- The constrained mechanisms we proposed for $\mathbb{L}_0$ are most effective for small datasets ($n \leq 20$). As n gets larger, GM starts exhibiting most properties.

- The LP mechanisms are optimal for a given loss function. Identifying use cases for each of the loss functions is not immediate and beyond the scope of our discussion. Ghosh *et al.*'s LP framework provides mechanisms optimal when the range and the domain are of equal size. As we have seen in the local case, one may be able to design more accurate mechanisms with this restriction relaxed.

Bibliography

[1] Dwork Cynthia. Reminiscences, 2018. URL DOI:https://doi.org/10.29012/jpc.702.

[2] URL https://en.wikipedia.org/wiki/List_of_countries_by_number_of_Internet_users.

[3] URL https://www.bbc.co.uk/iplayer.

[4] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pages 191–198, 2016. doi: 10.1145/2959100.2959190. URL https://doi.org/10.1145/2959100.2959190.

[5] Carlos A. Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4):13:1–13:19, December 2015. ISSN 2158-656X. doi: 10.1145/2843948. URL http://doi.acm.org/10.1145/2843948.

[6] Kurt Jacobson, Vidhya Murali, Edward Newett, Brian Whitman, and Romain Yon. Music personalization at spotify. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 373–373, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4035-9. doi: 10.1145/2959100.2959120. URL http://doi.acm.org/10.1145/2959100.2959120.

[7] Pablo Samuel Castro, Daqing Zhang, and Shijian Li. Urban traffic modelling and prediction using large scale taxi gps traces. In *Proceedings of the 10th International Conference on Pervasive Computing*, Pervasive'12, pages 57–72, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-31204-5. doi: 10.1007/978-3-642-31205-2_4. URL http://dx.doi.org/10.1007/978-3-642-31205-2_4.

[8] Henry Corrigan-Gibbs and Dan Boneh. Prio: Private, robust, and scalable computation of aggregate statistics. In *14th USENIX Symposium on Networked Systems Design*

and Implementation, NSDI 2017, Boston, MA, USA, March 27-29, 2017, pages 259–282, 2017. URL https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/corrigan-gibbs.

[9] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems 30*, December 2017. URL https://www.microsoft.com/en-us/research/publication/collecting-telemetry-data-privately/.

[10] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *ACM CCS*, 2014. ISBN 978-1-4503-2957-6. doi: 10.1145/2660267.2660348. URL http://doi.acm.org/10.1145/2660267.2660348.

[11] Abhishek Bhowmick, John Duchi1, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in privatefederated learning. URL https://arxiv.org/abs/1812.00984.

[12] Vasyl Pihur, Aleksandra Korolova, Frederick Liu, Subhash Sankuratripati, Moti Yung, Dachuan Huang, and Ruogu Zeng. Differentially-private "draw and discard" machine learning. *CoRR*, abs/1807.04369, 2018. URL http://arxiv.org/abs/1807.04369.

[13] URL https://en.wikipedia.org/wiki/Facebook\T1\textendashCambridge_Analytica_data_scandal.

[14] URL https://www.equifax.co.uk/incident.html.

[15] T. E. Dalenius. Towards a methodology for statistical disclosure control. 1977.

[16] Latanya Sweeney. Recommendations to identify and combat privacy problems in the commonwealth. URL http://dataprivacylab.org/dataprivacy/talks/Flick-05-10.html.

[17] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, 1998.

[18] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15-20, 2007*, pages 106–115, 2007. doi: 10.1109/ICDE.2007.367856. URL https://doi.org/10.1109/ICDE.2007.367856.

[19] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. *L*-diversity: Privacy beyond *k*-anonymity. *TKDD*, 1(1):3, 2007. doi: 10.1145/1217299.1217302. URL https://doi.org/10.1145/1217299.1217302.

[20] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov. "you might also like:" privacy risks of collaborative filtering. In *2011 IEEE Symposium on Security and Privacy*, pages 231–246, May 2011. doi: 10.1109/SP.2011.40.

[21] Arvind Narayanan, Elaine Shi, and Benjamin I. P. Rubinstein. Link prediction by de-anonymization: How we won the kaggle social network challenge. *CoRR*, abs/1102.4374, 2011. URL http://arxiv.org/abs/1102.4374.

[22] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *2009 30th IEEE Symposium on Security and Privacy*, pages 173–187, May 2009. doi: 10.1109/SP.2009.22.

[23] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Nature*, 2013. doi: https://doi.org/10.1038/srep01376. URL https://www.nature.com/articles/srep01376?ial=1.

[24] Philippe Golle and Kurt Partridge. On the anonymity of home/work location pairs. In *Pervasive Computing, 7th International Conference, Pervasive 2009, Nara, Japan, May 11-14, 2009. Proceedings*, pages 390–397, 2009. doi: 10.1007/978-3-642-01516-8_26. URL https://doi.org/10.1007/978-3-642-01516-8_26.

[25] Jessica Su, Ansh Shukla, Sharad Goel, and Arvind Narayanan. De-anonymizing web browsing data with social networks. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1261–1269, 2017. doi: 10.1145/3038912.3052714. URL https://doi.org/10.1145/3038912.3052714.

[26] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-32731-2, 978-3-540-32731-8. doi: 10.1007/11681878_14. URL http://dx.doi.org/10.1007/11681878_14.

[27] John M. Abowd. The U.S. census bureau adopts differential privacy. In *Proceedings of ACM SIGKDD*, pages 2867–2867. ACM, 2018. doi: 10.1145/3219819.3226070. URL http://doi.acm.org/10.1145/3219819.3226070.

[28] Differential Privacy Team, Apple. Learning with privacy at scale. 2017. URL https://machinelearning.apple.com/docs/learning-with-privacy-at-scale/appledifferentialprivacysystem.pdf.

[29] Giulia Fanti, Vasyl Pihur, and Úlfar Erlingsson. Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries. *Proceedings on Privacy Enhancing Technologies*, 2016(3):41–61, 2016.

[30] Noah M. Johnson, Joseph P. Near, and Dawn Xiaodong Song. Practical differential privacy for SQL queries using elastic sensitivity. *CoRR*, abs/1706.09479, 2017. URL http://arxiv.org/abs/1706.09479.

[31] Kobbi Nissim and Alexandra Wood. Is privacy privacy? *Philosophical Transaction of the Royal Society A*, 376(2128), 2018. URL http://rsta.royalsocietypublishing.org/content/376/2128/20170358.

[32] Aloni Cohen and Kobbi Nissim. Towards formalizing the gdpr's notion of singling out. *CoRR*, abs/1904.06009, 2019. URL http://arxiv.org/abs/1904.06009.

[33] Rachel Cummings and Deven Desai. The role of differential privacy in gdpr compliance, 2018. URL https://piret.gitlab.io/fatrec2018/program/fatrec2018-cummings.pdf.

[34] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. What can we learn privately? *SIAM J. Comput.*, 40(3): 793–826, 2011. doi: 10.1137/090756090. URL https://doi.org/10.1137/090756090.

[35] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *FOCS*. IEEE, 2013.

[36] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In *STOC*, 2009. ISBN 978-1-60558-506-2. doi: 10.1145/1536414.1536464. URL http://doi.acm.org/10.1145/1536414.1536464.

[37] Mangesh Gupte and Mukund Sundararajan. Universally optimal privacy mechanisms for minimax agents. In *PODS*, pages 135–146, 2010. ISBN 978-1-4503-0033-9. doi: 10.1145/1807085.1807105. URL http://doi.acm.org/10.1145/1807085.1807105.

[38] Aaron Schein, Zhiwei Steven Wu, Alexandra Schofield, Mingyuan Zhou, and Hanna M. Wallach. Locally private bayesian inference for count models. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 5638–5648, 2019. URL http://proceedings.mlr.press/v97/schein19a.html.

[39] Tejas Kulkarni, Graham Cormode, and Divesh Srivastava. Answering range queries under local differential privacy. *CoRR*, abs/1812.10942, 2018. URL http://arxiv.org/abs/1812.10942.

[40] Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. Marginal release under local differential privacy. *CoRR*, abs/1711.02952, 2017. URL https://dl.acm.org/citation.cfm?doid=3183713.3196906.

[41] *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018*, 2018. IEEE Computer Society. ISBN 978-1-5386-5520-7. URL http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8476188.

[42] Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. Constrained private mechanisms for count data. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2019. ISSN 1041-4347. doi: 10.1109/TKDE.2019.2912179.

[43] James Bell, Aurélien Bellet, Adrià Gascón, and Tejas Kulkarni. Private protocols for u-statistics in the local model and beyond, 2019.

[44] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice, 2018. URL ../papers/ldptutorial.pdf. Tutorial at SIGMOD and KDD.

[45] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2005.

[46] D. Jurafsky and J.H. Martin. *Speech and Language Processing*. Always learning. Pearson, 2014. ISBN 9781292025438. URL https://books.google.co.uk/books?id=km-kngEACAAJ.

[47] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, May 1968. ISSN 0018-9448. doi: 10.1109/TIT.1968.1054142.

[48] Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors. *Advances in Neural Information Processing Systems 13, Papers from*

Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA, 2001. MIT Press. URL http://papers.nips.cc/book/advances-in-neural-information-processing-systems-13-2000.

[49] Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography Conference*, 2006.

[50] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-32731-2, 978-3-540-32731-8. doi: 10.1007/11681878_14. URL http://dx.doi.org/10.1007/11681878_14.

[51] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, 2007. URL https://www.microsoft.com/en-us/research/publication/mechanism-design-via-differential-privacy/.

[52] Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: Simultaneously solving how and what. In *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, pages 451–468, 2008. doi: 10.1007/978-3-540-85174-5_25. URL https://doi.org/10.1007/978-3-540-85174-5_25.

[53] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Optimal lower bound for differentially private multi-party aggregation. In *Algorithms - ESA 2012 - 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings*, pages 277–288, 2012. doi: 10.1007/978-3-642-33090-2_25. URL https://doi.org/10.1007/978-3-642-33090-2_25.

[54] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, 2006.

[55] S. L. Warner. Randomised response: a survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, March 1965.

[56] A. Chaudhuri and R. Mukerjee. *Randomized response: Theory and techniques.* Marcel Dekker, 1988.

[57] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *26th USENIX*

Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017., pages 729–745, 2017. URL https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/wang-tianhao.

[58] Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. Marginal release under local differential privacy. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2018. URL https://arxiv.org/abs/1711.02952.

[59] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. In *Advances in neural information processing systems*, pages 2879–2887, 2014.

[60] Mark D. Flood, Jonathan Katz, Stephen J. Ong, and Adam Smith. Cryptography and the economics of supervisory information: balancing transparency and confidentiality. Working Papers (Old Series) 1312, Federal Reserve Bank of Cleveland, 2013. URL https://ideas.repec.org/p/fip/fedcwp/1312.html.

[61] Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 901–914, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2477-9. doi: 10.1145/2508859.2516735. URL http://doi.acm.org/10.1145/2508859.2516735.

[62] A. Agresti. *Categorical Data Analysis*. Wiley Series in Probability and Statistics. Wiley, 2003. ISBN 9780471458760. URL https://books.google.co.uk/books?id=hpEzw4T0sPUC.

[63] Alan Herschtal and Bhavani Raskutti. Optimising area under the roc curve using gradient descent. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 49–, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. doi: 10.1145/1015330.1015366. URL http://doi.acm.org/10.1145/1015330.1015366.

[64] David Evans, Vladimir Kolesnikov, and Mike Rosulek. A pragmatic introduction to secure multi-party computation. *Foundations and Trends® in Privacy and Security*, 2(2-3):70–246, 2018. ISSN 2474-1558. doi: 10.1561/3300000019. URL http://dx.doi.org/10.1561/3300000019.

[65] Trusted execution environment. URL https://en.wikipedia.org/wiki/Trusted_execution_environment.

[66] Alexandre V. Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, and Johannes Gehrke. Privacy preserving mining of association rules. In *ACM SIGKDD*, pages 217–228, 2002.

[67] Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Thakurta. Practical locally private heavy hitters. *CoRR*, abs/1707.04982, 2017. URL http://arxiv.org/abs/1707.04982.

[68] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *ACM CCS*, 2014. ISBN 978-1-4503-2957-6. doi: 10.1145/2660267.2660348. URL http://doi.acm.org/10.1145/2660267.2660348.

[69] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, July 1970. ISSN 0001-0782. doi: 10.1145/362686.362692. URL http://doi.acm.org/10.1145/362686.362692.

[70] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 715–724, 2010. doi: 10.1145/1806689.1806787. URL https://doi.org/10.1145/1806689.1806787.

[71] Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. "you might also like: " privacy risks of collaborative filtering. In *32nd IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA*, pages 231–246, 2011. doi: 10.1109/SP.2011.40. URL https://doi.org/10.1109/SP.2011.40.

[72] A Thakurta, A Vyrros, U Vaishampayan, G Kapoor, J Freudiger, V Rangarajan Sridhar, D Davidson. Private dictionary population satisfying local differential privacy, March 2017. URL http://pimg-fpiw.uspto.gov/fdd/41/947/095/0.pdf. US Patent 9,594,741 B1.

[73] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, ICALP '02, pages 693–703, Berlin, Heidelberg, 2002. Springer-Verlag. ISBN 3-540-43864-5. URL http://dl.acm.org/citation.cfm?id=646255.684566.

[74] Martin Farach-Colton, editor. *LATIN 2004: Theoretical Informatics, 6th Latin American Symposium, Buenos Aires, Argentina, April 5-8, 2004, Proceedings*, volume

2976 of *Lecture Notes in Computer Science*, 2004. Springer. ISBN 3-540-21258-2. doi: 10.1007/b95852. URL https://doi.org/10.1007/b95852.

[75] Matthew Joseph, Aaron Roth, Jonathan Ullman, and Bo Waggoner. Local differential privacy for evolving data. February 2018.

[76] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. *CoRR*, abs/1811.12469, 2018. URL http://arxiv.org/abs/1811.12469.

[77] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and XiaoFeng Wang. Privacy loss in apple's implementation of differential privacy on macos 10.12. *CoRR*, abs/1709.02753, 2017. URL http://arxiv.org/abs/1709.02753.

[78] Raef Bassily and Adam Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of ACM STOC*, pages 127–135. ACM, 2015.

[79] Mark Bun, Jelani Nelson, and Uri Stemmer. Heavy hitters and the structure of local privacy. *CoRR*, abs/1711.04740, 2017.

[80] T. Wang, N. Li, and S. Jha. Locally differentially private frequent itemset mining. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 127–143, May 2018. doi: 10.1109/SP.2018.00035.

[81] Zhan Qin, Yin Yang, Ting Yu, Issa Khalil, Xiaokui Xiao, and Kui Ren. Heavy hitter estimation over set-valued data with local differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 192–203. ACM, 2016.

[82] Jinyuan Jia and Neil Zhenqiang Gong. Calibrate: Frequency estimation and heavy hitter identification with local differential privacy via incorporating prior knowledge. *CoRR*, abs/1812.02055, 2018. URL http://arxiv.org/abs/1812.02055.

[83] Zhan Qin, Ting Yu, Yin Yang, Issa Khalil, Xiaokui Xiao, and Kui Ren. Generating synthetic decentralized social graphs with local differential privacy. In *CCS*, pages 425–438. ACM, 2017.

[84] Rui Chen, Haoran Li, A. K. Qin, Shiva Prasad Kasiviswanathan, and Hongxia Jin. Private spatial data aggregation in the local setting. In *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*, pages 289–300, 2016. doi: 10.1109/ICDE.2016.7498248. URL https://doi.org/10.1109/ICDE.2016.7498248.

[85] Thông T. Nguyên, Xiaokui Xiao, Yin Yang, Siu Cheung Hui, Hyejin Shin, and Junbum Shin. Collecting and analyzing data from smart device users with local differential privacy. *CoRR*, abs/1606.05053, 2016. URL http://arxiv.org/abs/1606.05053.

[86] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. Collecting and analyzing multidimensional data with local differential privacy. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*, pages 638–649, 2019. doi: 10.1109/ICDE.2019.00063. URL https://doi.org/10.1109/ICDE.2019.00063.

[87] Teng Wang, Jun Zhao, Xinyu Yang, and Xuebin Ren. Locally differentially private data collection and analysis. 2019. URL https://arxiv.org/abs/1906.01777.

[88] Hyejin Shin, Sungwook Kim, Junbum Shin, and Xiaokui Xiao. Privacy enhanced matrix factorization for recommendation with local differential privacy. *IEEE Trans. Knowl. Data Eng.*, 30(9):1770–1782, 2018. doi: 10.1109/TKDE.2018.2805356. URL https://doi.org/10.1109/TKDE.2018.2805356.

[89] Haipei Sun, Boxiang Dong, Wendy Hui Wang, Ting Yu, and Zhan Qin. Truth inference on sparse crowdsourcing data with local differential privacy. In *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*, pages 488–497, 2018. doi: 10.1109/BigData.2018.8622635. URL https://doi.org/10.1109/BigData.2018.8622635.

[90] Adam D. Smith, Abhradeep Thakurta, and Jalaj Upadhyay. Is interaction necessary for distributed private learning? In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 58–77, 2017. doi: 10.1109/SP.2017.35. URL https://doi.org/10.1109/SP.2017.35.

[91] Kai Zheng, Wenlong Mou, and Liwei Wang. Collect at once, use effectively: Making non-interactive locally private learning possible. *CoRR*, abs/1706.03316, 2017. URL http://arxiv.org/abs/1706.03316.

[92] Di Wang, Marco Gaboardi, and Jinhui Xu. Empirical risk minimization in non-interactive local differential privacy revisited. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 965–974. Curran Associates, Inc., 2018. URL https://pdfs.semanticscholar.org/c6be/7b30c1e9a0a3d49642d40a52b62f1ec6dd20.pdf.

[93] Di Wang, Adam Smith, and Jinhui Xu. Differentially private empirical risk minimization in non-interactive local model via polynomial of inner product approximation. *CoRR*, abs/1812.06825, 2018. URL http://arxiv.org/abs/1812.06825.

[94] Di Wang and Jinhui Xu. On sparse linear regression in the local differential privacy model. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 6628–6637, 2019. URL http://proceedings.mlr.press/v97/wang19m.html.

[95] Kobbi Nissim and Uri Stemmer. Clustering algorithms for the centralized and local models. In *Algorithmic Learning Theory, ALT 2018, 7-9 April 2018, Lanzarote, Canary Islands, Spain*, pages 619–653, 2018. URL http://proceedings.mlr.press/v83/nissim18a.html.

[96] Kobbi Nissim and Uri Stemmer. Clustering algorithms for the centralized and local models. *CoRR*, abs/1707.04766, 2017. URL http://arxiv.org/abs/1707.04766.

[97] Ning Wang, Xiaokui Xiao, Yin Yang, Ta Duy Hoang, Hyejin Shin, Junbum Shin, and Ge Yu. Privtrie: Effective frequent term discovery under local differential privacy. In *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018*, pages 821–832, 2018. doi: 10.1109/ICDE.2018.00079. URL https://doi.org/10.1109/ICDE.2018.00079.

[98] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnés, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*, pages 441–459, 2017. doi: 10.1145/3132747.3132769. URL https://doi.org/10.1145/3132747.3132769.

[99] Albert Cheu, Adam D. Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via mixnets. *CoRR*, abs/1808.01394, 2018. URL http://arxiv.org/abs/1808.01394.

[100] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. *CoRR*, abs/1903.02837, 2019. URL http://arxiv.org/abs/1903.02837.

[101] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency, 2016.

[102] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning, 2019.

[103] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private language models without losing accuracy. *CoRR*, abs/1710.06963, 2017. URL `http://arxiv.org/abs/1710.06963`.

[104] H. Brendan McMahan and Galen Andrew. A general approach to adding differential privacy to iterative training procedures. *CoRR*, abs/1812.06210, 2018. URL `http://arxiv.org/abs/1812.06210`.

[105] Wennan Zhu, Peter Kairouz, Haicheng Sun, Brendan McMahan, and Wei Li. Federated heavy hitters with differential privacy, 2019.

[106] Brendan Avent, Aleksandra Korolova, David Zeber, Torgeir Hovden, and Benjamin Livshits. BLENDER: enabling local search with a hybrid differential privacy model. In *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017.*, pages 747–764, 2017. URL `https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/avent`.

[107] Yatharth Dubey and Aleksandra Korolova. The power of the hybrid model for mean estimation. *CoRR*, abs/1811.12040, 2018. URL `http://arxiv.org/abs/1811.12040`.

[108] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms. In *ICDE*, 2010.

[109] 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 *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 273–282. ACM, 2007.

[110] Bolin Ding, Marianne Winslett, Jiawei Han, and Zhenhui Li. Differentially private data cubes: Optimizing noise sources and consistency. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 217–228, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0661-4. doi: 10.1145/1989323.1989347. URL http://doi.acm.org/10.1145/1989323.1989347.

[111] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'12, pages 2339–2347, USA, 2012. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=2999325.2999396.

[112] Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 61–70. IEEE, 2010.

[113] Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. Privately releasing conjunctions and the statistical query barrier. *SIAM Journal on Computing*, 42(4): 1494–1520, 2013.

[114] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 94–103, 2007. doi: 10.1109/FOCS.2007.41. URL https://doi.org/10.1109/FOCS.2007.41.

[115] Graham Cormode, Minos Garofalakis, Peter Haas, and Chris Jermaine. *Synposes for Massive Data: Samples, Histograms, Wavelets and Sketches*. Foundations and Trends in Databases. *NOW publishers*, 2012.

[116] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems*, pages 2348–2356, 2012.

[117] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms. *IEEE Trans. Knowl. Data Eng.*, 23(8):1200–1214, 2011. doi: 10.1109/TKDE.2010.247. URL https://doi.org/10.1109/TKDE.2010.247.

[118] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow.*, 3(1-2): 1021–1032, September 2010. ISSN 2150-8097. doi: 10.14778/1920841.1920970. URL http://dx.doi.org/10.14778/1920841.1920970.

[119] Wahbeh Qardaji, Weining Yang, and Ninghui Li. Understanding hierarchical methods for differentially private histograms. *Proc. VLDB Endow.*, 6(14):1954–1965, September 2013. doi: 10.14778/2556549.2556576.

[120] Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. Differentially private spatial decompositions. In *IEEE 28th International Conference on Data Engineering*, pages 20–31, 2012.

[121] Wahbeh H. Qardaji, Weining Yang, and Ninghui Li. Differentially private grids for geospatial data. In *Proceedings of IEEE ICDE*, pages 757–768, 2013.

[122] Hai Brenner and Kobbi Nissim. Impossibility of differentially private universally optimal mechanisms. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 71–80, 2010.

[123] Justin Thaler, Jonathan Ullman, and Salil Vadhan. Faster algorithms for privately releasing marginals. In *International Colloquium on Automata, Languages, and Programming*, pages 810–821. Springer, 2012.

[124] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, New York, NY, USA, 2014. ISBN 1107038324, 9781107038325.

[125] Kulkarni Tejas. Answering marginal queries under ldp, 2017. URL `https://gitlab.com/Tejasvk/MarginalsCode`.

[126] NYC taxi and limousine commission, trip record data, 2017. URL `http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml`.

[127] N. Lockyer. *Nature*. Number v. 15. Macmillan Journals Limited, 1877. URL `https://books.google.fi/books?id=eskKAAAAYAAJ`.

[128] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, December 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL `http://doi.acm.org/10.1145/2827872`.

[129] Thông T. Nguyên, Xiaokui Xiao, Yin Yang, Siu Cheung Hui, Hyejin Shin, and Junbum Shin. Collecting and analyzing data from smart device users with local differential privacy. *CoRR*, abs/1606.05053, 2016. URL `http://arxiv.org/abs/1606.05053`.

[130] Marco Gaboardi, Ryan M Rogers, and Salil P Vadhan. Differentially private chi-squared hypothesis testing: Goodness of fit and independence testing. 2016.

[131] Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. Marginal release under local differential privacy. In *Proceedings of ACM SIGMOD*, pages 131–146, 2018.

[132] Zhikun Zhang, Tianhao Wang, Ninghui Li, Shibo He, and Jiming Chen. CALM: consistent adaptive local marginal for marginal release under local differential privacy. In *Proceedings of ACM CCS 2018*, pages 212–229, 2018.

[133] Wahbeh Qardaji, Weining Yang, and Ninghui Li. Priview: Practical differentially private release of marginal contingency tables. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 1435–1446, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2376-5. doi: 10.1145/2588555.2588575. URL http://doi.acm.org/10.1145/2588555.2588575.

[134] Di Wang, Marco Gaboardi, and Jinhui Xu. Efficient empirical risk minimization with smooth loss functions in non-interactive local differential privacy. *CoRR*, abs/1802.04085, 2018. URL http://arxiv.org/abs/1802.04085.

[135] J.C. Mason and D.C. Handscomb. *Chebyshev Polynomials*. CRC Press, 2002. ISBN 9781420036114. URL https://books.google.co.uk/books?id=8FHf0P3to0UC.

[136] Ashwin Paranjape, Austin R. Benson, and Jure Leskovec. Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 601–610, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4675-7. doi: 10.1145/3018661.3018731. URL http://doi.acm.org/10.1145/3018661.3018731.

[137] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD*, 2011.

[138] https://en.wikipedia.org/wiki/Geohash. URL https://en.wikipedia.org/wiki/Geohash.

[139] Kulkarni Tejas. Answering range queries under ldp, 2018. URL https://gitlab.com/Tejasvk/RangeQueries.

[140] Q. Geng, P. Kairouz, S. Oh, and P. Viswanath. The staircase mechanism in differential privacy. *IEEE Journal of Selected Topics in Signal Processing*, 9(7):1176–1184, Oct 2015. ISSN 1932-4553. doi: 10.1109/JSTSP.2015.2425831.

[141] pylpsolve. Pylpsolve- an object oriented wrapper for the lpsolve. www.stat.washington.edu/~hoytak/code/pylpsolve/, 2010.

[142] C. Blake and C. Merz. UCI machine learning repository, 1998. URL `https://archive.ics.uci.edu/ml/datasets/Adult`.

[143] Bradley Efron and Charles Stein. The jackknife estimate of variance. *The Annals of Statistics*, pages 586–596, 1981.