Marginal Release Under Local Differential Privacy

Graham Cormode*
University Of Warwick, UK
g.cormode@warwick.ac.uk

Tejas Kulkarni†
University Of Warwick, UK
tejasvijaykulkarni@gmail.com

Divesh Srivastava
AT&T Labs-Research, USA
divesh@research.att.com

ABSTRACT

Many analysis and machine learning tasks require the availability of marginal statistics on multidimensional datasets while providing strong privacy guarantees for the data subjects. Applications for these statistics range from finding correlations in the data to fitting sophisticated prediction models. In this paper, we provide a set of algorithms for materializing marginal statistics under the strong model of local differential privacy. 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.

ACM Reference Format:
Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. 2018. Marginal Release Under Local Differential Privacy. In *Proceedings of 2018 International Conference on Management of Data (SIGMOD'18).* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3183713.3196906

1 INTRODUCTION

Modern data-driven applications must guarantee a high level of privacy to their users if they are to gain widespread acceptance. The current de facto standard for privacy is differential privacy, which imposes a statistical requirement on the output of a data release process. Considerable effort has been invested into achieving this guarantee while maximizing the fidelity of the released information, typically with the assistance of a centralized trusted third party who aggregates the data. However, there is growing importance placed on algorithms which dispense with the trusted aggregator, and instead allow each participant to ensure that the information that they reveal already meets the differential privacy guarantee in isolation. This gives the *local differential privacy* (LDP) model.

The model of Local Differential Privacy combines the statistical guarantees of differential privacy with a further promise to the user: their information is never visible to anyone else in its raw form, and they retain "plausible deniability" of any sensitive information associated with them. Local differential privacy has been adopted in Google Chrome via the RAPPOR tool to collect browing and system statistics [18], and in Apple's iOS 10 to collect app usage statistics [2]. It is consequently already deployed for information gathering over a user base of hundreds of millions of people. So far, work in the LDP model has focused on relatively simple computations: collecting basic frequency statistics about one dimensional data (e.g. identifying popular destinations on the web). But this one-dimensional view of the world does not capture the rich set of correlations that real data exhibits, as in the following case study.

Motivating example: movement patterns. Consider the collection and release of statistics on movement patterns of individual's. Each trip is a multidimensional object, including origin and destination, timings, tip, and mode of payment. All of these should be considered private, as they can potentially be used to determine an individual's work/home locations, social habits and lifestyle. Figure 1 shows an example taxi trip dataset, where each journey is described in terms of a number of (binary) attributes. "Anonymized" taxi trip data from New York City still made it possible to stalk celebrities,

Table 1: Attributes of NYC taxi dataset

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

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

Figure 1: Trip Data

M_pick/M_drop	Y	N
Y	0.55	0.15
N	0.10	0.20

Figure 2: 2-way marginal

*Supported in part by European Research Council grant ERC-2014-CoG 647557 and The Alan Turing Institute under the EPSRC grant EP/N510129/1.
†Supported in part by AT&T, Marie Curie CIG 618202 and The Alan Turing Institute under the EPSRC grant EP/N510129/1.

SIGMOD'18, June 10–15, 2018, Houston, TX, USA

ACM ISBN 978-1-4503-4703-7/18/06…$15.00
https://doi.org/10.1145/3183713.3196906

identify drivers doing good business, and infer home/work locations of frequent visitors to adult entertainment clubs. However, there are many positive potential uses of this type of data when it is represented as statistical marginal tables. Put simply, a marginal table records the (empirical) probability distribution between a set of attributes. A taxi service provider could learn more about the travel needs, habits and preferences of customers in a particular market, e.g. which areas are most popular during the night versus during the day. We would want to test the validity of any correlations found by subjecting them to statistical hypothesis tests. Beyond just finding correlations, we may be interested in more complex analysis such as describing the probabilistic relationship between cause and effects by modeling features as a Bayesian Network. This could help to predict demand in different areas at different times, reduce customer waiting times, and provide expected revenues. It could even infer customer satisfaction with different drivers by comparing average tip amounts across collections of comparable journeys. Such graphical models rely on the computation of conditional probability tables (CPTs), which are derived directly from marginals. All these tasks can be accomplished given information about the population-level correlations between particular combinations of attributes, which can be derived from the marginal distributions between small numbers of attributes–typically, two-way or three-way marginals suffice. Figure 2 shows a marginal table 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. Consequently, our objective is to allow such tables to be computed accurately on demand from data collected privately from a large user population. □

Thus, the contingency, or marginal, table is "the workhorse of data analysis" [4]. 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 [26] and association rule mining (market basket analysis) compute low order marginals as a preprocessing step. Furthermore, for multivariate distributions where direct sampling is infeasible or too costly, low dimensional marginals serve as building blocks [9, 29] to compute accurate approximations. Hence, we identify low-order (2-way and 3-way) marginals as our main focus.

It is therefore not surprising that much work on private data analysis has studied the problem of materializing and releasing marginals while achieving some privacy guarantee. It is clear that the information described by marginals is potentially very sensitive, as it collates and reveals information about individuals. A canonical example of privacy leakage is when a cell in a marginal table refers to just one person or a few individuals, and allows the value of an attribute to be inferred. For instance, a marginal table relating salary and zip code can reveal an individual's income level when they are the sole high earner in a region. While this problem has been well-studied under centralized differential privacy, when we desire to protect individuals under the stronger model of local DP, there is limited prior work ([20] for 2-way marginals); in this paper, we seek to give strong guarantees for arbitrary marginals.

Our Contributions. In this work, we provide a general framework for marginal release under LDP, with theoretical and empirical analysis. First, we review prior work (Section 2), and provide background on methods to support private marginal release (Section 3). We 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). We perform experimental comparison to augment the theoretical understanding in Section 5, 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 favor in the centralized model, supplanted by more involved privacy mechanisms [12, 23, 39]. We observe that these other mechanisms do not easily translate to the local model. Concurrent with the development of this work, the Hadamard basis has found application in protocols for LDP frequency estimation [10]. 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. We summarize the results in Table 2. Our methods are eminently suitable for implementation in existing LDP deployments (Chrome, iOS) for gathering correlation statistics.

2 RELATED WORK

Differential Privacy (DP) [13, 15, 16], unlike its precursor privacy definitions, provides semantic mathematical promises on individuals' privacy, interpreted as statistical properties of the output distribution of a randomized algorithm. Formally, an algorithm $\mathcal{A}$ meets the ϵ-DP guarantee if, over pairs of inputs D, D' that differ in the presence of a single individual, its output $\mathcal{A}(D)$ satisfies

$$\frac{\Pr[\mathcal{A}(D) = x]}{\Pr[\mathcal{A}(D') = x]} \leq \exp(\epsilon), \tag{1}$$

where x is any permitted output. The model has risen in popularity and adoption compared to earlier attempts to codify privacy (such as k-anonymity and ℓ-diversity [30, 37]), which can leak sensitive information. It has been a topic of inquiry for diverse research communities including theory [40], data management [31], machine learning [38], and systems/programming languages [33]. We focus on two directions: the emergent model of *local* differential privacy (Section 2.1), and private marginal release (Section 2.2).

2.1 Local Differential Privacy (LDP)

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. This model was first suggested by Evfimievski *et al.* [19] under the name of γ-amplification, with an application to mining association rules. Duchi *et al.* [14] studied a generalization of that model as a local version of DP, and proposed a minimax framework with information theoretic bounds on utility.

The canonical LDP algorithm is randomized response (RR), first developed in the context of survey design in the 1960s [42]. Users who possess a private bit of information flip it with some probability p to have plausible deniability of their response. Collecting enough responses shared through this mechanism allows an accurate estimate of the population behavior to be formed. Randomized response is at the heart of many recent LDP algorithms, most famously Google's deployment of RAPPOR [18], where RR is applied to a Bloom filter encoding a large set of possible URLs. In a follow-up paper, Fanti *et al.* [20] extend RAPPOR's ability to identify strings which are frequent across the user distribution, building them up piece by piece. However, their solution is somewhat specific to RAPPOR's case and does not offer any guarantee on the accuracy. Apple's implementation uses sketching to reduce the dimensionality of a massive domain [2, 5, 10]. Microsoft telemetry data collection involves rounding and memoization technique for periodic measurement of app usage statistics under LDP [11].

There is a growing theoretical understanding of LDP. Kairouz *et al.* [27, 28] study how to estimate the frequency of a single categorical attribute, and propose optimal generalizations of randomized response. Closest to our interest is work on generating a histogram under LDP, or identifying the peaks in the input (heavy hitters). This can be viewed as the problem of estimating a one-dimensional marginal distribution. The state of art asymptotic lower bound and matching algorithm are due to Bassily and Smith [6]. They adapt ideas from dimensionality reduction (i.e. the Johnson-Lindenstrauss lemma) to build a primitive to estimate the weight of a single point in the distribution; this is used to find all heavy hitters. Qin *et al.* adapt this approach to the related problem of identifying heavy hitters within set valued data [36]; Chen *et al.* use it on spatial data to build user movement models under LDP [8]. Wang *et al.* describe optimizations that consider asymmetric randomized response and hashing to reduce variance [41]. We apply some of these methods in our setting in Appendix B.2, but find that they do not scale to the range of input dimensions that we consider. Nguyên *et al.* describe a general approach for data analysis under LDP with multiple rounds [34]. They propose an orthogonal measurement basis isomorphic to the Hadamard transform, but only for 1D data.

2.2 Publishing Marginals Privately

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, these all 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, and 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 $O(1/N)$ [15]. A variant is to apply this to a transformation of the data, such as a wavelet or Fourier transform [4, 43]. 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. [12] 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 [22–24]. For concreteness, we describe a non-adaptive approach due to Hardt *et al.* [23]. 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 [32] 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 rescaling 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.

Chebyshev polynomials. Thaler *et al.* view a dataset as a linear function on marginals, and represent each record of a dataset via a γ-accurate Chebyshev polynomial [39], whose coefficients can be perturbed for privacy. This is intended to be faster than multiplicative weights solutions which must range over an $O(2^d)$-sized representation of the data. This approach could plausibly be adapted to LDP, although the steps required are far from immediate.

In summary, the LDP requirement to perturb elements of every single record independently (which are sparse in our case) while

preserving the underlying correlations is not yet met by prior work, and so we must give new algorithms and analyses.

3 MODEL AND PRELIMINARIES

In line with prior work [4], our main focus is on data represented by binary variables. This helps to keep the notation uniform, and highlights the key challenges. We discuss the modifications necessary to accommodate more general attributes in Section 7.

In our setting, each user i has a 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 t_i of length 2^d with 1 at exactly one place j_i and 0's at remaining positions. The domain of all such t_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. We now formalize Local Differential Privacy (Section 3.1), and introduce examples and notation for computing marginals (Section 3.2).

3.1 Local Differential Privacy

Local differential privacy (LDP) requires each data owner to perturb their output to meet the DP guarantee. Any two tuples $t_i, t_i' \in \mathcal{I}_{2^d \times 2^d}$ are considered *adjacent*, with $||t_i - t_i'||_1 = 2$ i.e. t_i, t_i' are adjacent if they differ in their positions of 1's. LDP upper bounds the ratio of probabilities of seeing the same outcome for all adjacent tuples. The definition is obtained by applying the differential privacy definition of equation (1) to a single user's data.

Definition 3.1 (ϵ-local differential privacy (LDP) [14]). A randomized mechanism F is differentially private with parameter ϵ iff for all pairs of adjacent input tuples $t_i, t_i' \in \mathcal{I}_{2^d \times 2^d}$, and every possible output tuple R ($k \leq d$), we have

$$\Pr[F(t_i) = R] \leq e^{\epsilon} \Pr[F(t_i') = R] \tag{2}$$

When we aggregate in the LDP model, the above definition ensures that we cannot confidently distinguish whether R is an outcome of $F(t_i)$ or $F(t_i')$, yielding plausible deniability to user i. Note that in LDP each user does reveal their presence in the input. The model allows each user to operate with a different privacy parameter, but for simplicity we state our results using a value of ϵ which is shared by all users (in common with other work on LDP).

Basic Private Mechanisms. We describe primitives for LDP on simple inputs, which form building blocks for our protocols.

Randomized Response (RR): We first formally define the classic mechanism for releasing a single bit b_i under privacy by having the user lie with some probability [7]. In its simplest form, randomized response has each user i report the true value of their input (b_i) with probability $p_r > \frac{1}{2}$. Otherwise, i gives the opposite response ($1 - b_i$). It is immediate that RR admits differential privacy with $e^{\epsilon} = \frac{p_r}{1-p_r}$, by considering the probabilities of the four combinations of input and output. Its simplicity has made it popular in practical systems [2, 18, 34].

Budget Splitting (BS) and Randomized Response with Sampling (RRS): When each user holds m pieces of information, a first approach is to release information about all m via a mechanism that achieves (ϵ/m)-LDP on each, thus effectively splitting the "privacy budget" ϵ (BS). Standard composition results from the DP literature ensure that BS meets ϵ-LDP [15]. However, in general, accuracy is improved if we instead sample one out of m pieces of information and release this with ϵ-LDP [6], and this is confirmed by our analysis of our protocols. In particular, if a user's information is represented as a binary vector of dimension m, we can uniformly sample an index j with probability $p_s = 1/m$, and use Randomized Response with parameter p_r to release the value found there.

We often encounter cases where a user holds a sparse vector: exactly one entry is 1, and the rest are 0. The random sampling approach applied to the entries of the vector has the disadvantage that most likely we will sample a zero entry, limiting the information revealed. We discuss two alternative approaches, which extend randomized response in different ways.

Preferential Sampling (PS): The first extension of RR is a natural generalization (aka Generalized Randomized Reponse [28], Direct Encoding [41]), which we label Preferential Sampling. Given a sparse vector $t \in \mathcal{I}_{2^m \times 2^m}$ such that $|t| = 1$ and $t[j] = 1$, we sample an index ℓ according to the following distribution:

$$\ell = \begin{cases} j : t[j] = 1, & \text{with probability } p_s \\ j' : j' \in_R [m] \setminus \{j\}, & \text{with probability } 1 - p_s \end{cases}$$

In other words, we report the true index with probability p_s, while each incorrect index is reported with probability $\frac{1-p_s}{m-1}$. When $m = 2$ this mechanism is equivalent to 1 bit randomized response. Considering these two output probabilities, we immediately have:

Fact 3.1. *The output of preferential sampling meets LDP with* $e^{\epsilon} = \frac{p_s}{1-p_s} \cdot m - 1$.

Rearranging, we set $p_s = (1 + (m-1)e^{-\epsilon})^{-1}$ to obtain ϵ-LDP.

Parallel Randomized Response (PRR): A second approach is apply m independent instances of RR, each with parameter $\epsilon/2$. We refer to this as Parallel Randomized response (PRR) (also known as BasicRAPPOR [18] and Unary Encoding [41]). Note that the output of PRR is an m-bit string which is not guaranteed to be sparse.

Fact 3.2. *Parallel Randomized Reponse applied to a sparse vector* t *meets* ϵ*-LDP.*

This fact follows by observing that for adjacent inputs t_i and t_i', and a particular output R in (2), the probabilities associated with all but two output bit locations in R are identical (and so cancel in the ratio $\Pr[F(t_i) = R]/\Pr[F(t_i')] = R$). We are left with the probabilities associated with the locations j_i and j_i' (i.e. the locations 1 bits in the two adjacent inputs). The probability ratio for each of these bits is $\exp(\epsilon/2)$ from $\frac{\epsilon}{2}$-RR, so their product is $\exp(\epsilon)$, as required by (2). In

recent work, Wang et al. [41] set the probability of keeping the sole 1 to be $\frac{1}{2}$, and the probability of retaining each 0 to be $(1+e^{\epsilon})^{-1}$, to slightly improve the variance of this mechanism.

3.2 Marginal Tables

Notation and preliminaries. Recall that we model each user i's bit vector $t_i \in \mathcal{I}_{2^d \times 2^d}$ as a vertex in a d-dimensional Hamming cube. 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 3.2 (Marginal operator). Given a vector $t \in \mathbb{R}^{2^d}$, the *marginal operator* $C^{\beta} : \mathbb{R}^{2^d} \Rightarrow \mathbb{R}^{2^k}$ computes the summed frequencies for all combinations of values of attributes encoded by $\beta \in \{0,1\}^d$, where $|\beta|$, the number of 1s 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 $C^{0101}(t)$ 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 $C^{0101}(t)$ 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 $C^{\beta}(t)$ 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:

$$\forall \gamma \preceq \beta \quad C^{\beta}(t)[\gamma] = \textstyle\sum_{\eta : \eta \wedge \beta = \gamma} t[\eta] \tag{3}$$

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

Example 3.1. *Let $d = 4$ and $\beta = 0101$. Then, applying* (3)*:*
$C^{0101}(t)[\mathbf{0000}] = t[\mathbf{0000}] + t[\mathbf{0010}] + t[\mathbf{1000}] + t[\mathbf{1010}]$
$C^{0101}(t)[\mathbf{0001}] = t[\mathbf{0001}] + t[\mathbf{0011}] + t[\mathbf{1001}] + t[\mathbf{1011}]$
$C^{0101}(t)[\mathbf{0100}] = t[\mathbf{0100}] + t[\mathbf{0110}] + t[\mathbf{1100}] + t[\mathbf{1110}]$
$C^{0101}(t)[\mathbf{0101}] = t[\mathbf{0101}] + t[\mathbf{0111}] + t[\mathbf{1101}] + t[\mathbf{1111}]$
All indices in $\{0,1\}^d$ contribute exactly once to one entry in C^{0101}.

Definition 3.3 (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, $\forall j \leq k$.

Note that the (unique) d-way marginal corresponds to the complete input distribution. Since a single user's input t_i is sparse i.e. contains just a single 1 (say at index j_i), any marginal β of t_i will also be sparse with just one non-zero element. The relevant index in $C^{\beta}(t_i)$ is given by the bitwise operation $j_i \wedge \beta$.

Definition 3.4 (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 $t = \sum_{i=1}^{N} t_i/N$. Let $\widehat{C^{\beta}}$ be the approximate answer. We measure the quality of this in terms of the *total variation distance* from the true answer $C^{\beta}(t)$, i.e.

$$\tfrac{1}{2} \textstyle\sum_{\gamma \preceq \beta} |\widehat{C^{\beta}}[\gamma] - C^{\beta}(t)[\gamma]| = \tfrac{1}{2} \|\widehat{C^{\beta}} - C^{\beta}(t)\|_1$$

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 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 $\tilde{O}$ 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 3.5 (Hadamard Transformation (HT)). The transform of vector $t \in \mathbb{R}^{2^d}$ is $\theta = \phi t$ 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 $\mathbb{R}^{2^d}$. Given an arbitrary vector t, we say that its representation under the HT is given by the 2^d *Hadamard coefficients* (denoted as θ) in the vector $\theta = \phi t$. These properties of HT are well-known due to its role in the theory of Boolean functions [35]. In our case when t_i has only a single 1 (say at index ℓ), the Hadamard transform of t_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:

Lemma 3.6. $\phi(\sum_{i=1}^{n} t_i/N) = \frac{1}{N} \sum_{i=1}^{n} (\phi t_i)$

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. [4] is that we can write any marginal $\beta \in C$ as a sum of only a few Hadamard coefficients.

Lemma 3.7 ([4]). *Hadamard coefficients $H_k = \{\theta_\alpha : |\alpha| \leq k\}$ are sufficient to evaluate any k-way marginal β. Specifically,*

$$C^{\beta}(t)_\gamma = \sum_{\alpha \preceq \beta} \langle \phi_\alpha, t \rangle \sum_{\eta : \eta \wedge \beta = \gamma} \phi_{\alpha,\eta} = \sum_{\alpha \preceq \beta} \theta_\alpha \Big(\sum_{\eta : \eta \wedge \beta = \gamma} \phi_{\alpha,\eta} \Big) \tag{4}$$

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

Considering Example 3.1, 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 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 3.1, when the user's data is represented as a sparse input vector, we can instantiate this by sampling a cell in their table, and applying Randomized Response (the randomized response with sampling, RRS, approach); by reporting a single cell index (via the preferential sampling approach (PS)); or parallel randomized response (PRR) to report information on the vector. 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). Note that it is not meaningful to apply preferential sampling or parallel RR after Hadamard transform, since the input no longer meets the necessary sparsity assumption.

4.1 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 4.1 (Bernstein and Hoeffding inequalities). Given N independent variables X_i such that $\mathsf{E}[X_i] = 0$, $|X_i| < M_i$, and $\mathsf{Var}[X_i] = \sigma^2$ for all i. Then for any $c > 0$,

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

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 $\sigma^2 < M = \max_i M_i$.

We will show that our combination of techniques provides results which are unbiased, accurate, and private to estimate a parameter of the population f, such that $0 \leq f \leq 1$, using N observations.

Constructing unbiased estimators. It is straightforward to derive versions of the baseline operators (RRS, PS etc.) which are unbiased – that is, we derive values such that simple summation of these values provides a random value whose expectation is the sum of the input values. For concreteness, we describe the process for Preferential sampling to illustrate this step.

Recall that preferential sampling reports an index ℓ which is claimed to be the location of the sole 1 in the given input table of size 2^k. For convenience, we write $D = 2^k - 1$. Given the N reports, let F_j be the fraction of times that j is reported, and let f_j be the true fraction of inputs that have a 1 at location j. Then the expected reports of j come from the proportion of times the input is j and it is correctly reported ($p_s f_j$), plus the proportion of times the input is not j but j is chosen to be reported ($(1 - f_j)\frac{1-p_s}{D}$). Thus

$$\mathsf{E}[F_j] = p_s f_j + (1 - p_s)\frac{1-f_j}{D}$$

Rearranging this provides an unbiased estimator, $\hat{f}_j = \frac{DF_j + p_s - 1}{Dp_s + p_s - 1}$.

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

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

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. For ease of reading, we defer all proofs to the Appendix.

4.2 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 $\tilde{O}(N)$, 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 $t_i \in \mathcal{I}_{2^d \times 2^d}$.

Parallel Randomized Response On Input (InpRR). The most direct application of LDP here is to add noise to all 2^d locations.

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

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

Algorithm 1 User's routine for InpHT

procedure InpHT(t_i)
  Let $j_i \in \{0,1\}^d$ be the signal index of $t_i \in \mathcal{I}_{2^d \times 2^d}$.
  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**
  $\Theta^*[j] \leftarrow 2^{-d} \frac{\sum_{i=1}^{N} H_i^*[j]/(2p-1)}{N_j}$. ▷ N_j is the frequency count of index j.

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 4.3. *InpRR achieves ϵ-LDP and guarantees that $\|C^\beta(t) - C^\beta(t^*)\|_1 = \tilde{O}\left(\frac{2^{(d+k)/2}}{\epsilon\sqrt{N}}\right)$ with constant probability.*

Preferential Sampling On Input (InpPS). Our second method uses preferential sampling 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 t^* is found by applying the unbiasing to each noisy report (Section 4.1), and computing the average. As in the previous case, we can obtain any desired marginal by aggregating the reconstructed distribution.

Theorem 4.4. *InpPS achieves ϵ-LDP and guarantees that with constant probability we have for a target k-way marginal β*

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

Consequently, we get a guarantee for InpPS in terms of total variation distance of $\tilde{O}\left(\frac{2^{k/2}2^d}{\epsilon\sqrt{N}}\right)$. 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 3.7, 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 $|T| = \sum_{\ell=1}^{k} \binom{d}{\ell} = O(d^k)$ 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 rescales 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 3.7 to generate $C^\beta(t^*)$.

For completeness, Algorithms 1 and 2 spell out the transformations 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 4.2 to this setting to bound the total variation distance between true and reconstructed marginals.

Theorem 4.5. *InpHT achieves ϵ-LDP, and with constant probability we have for any target k-way marginal β*

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

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 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. For brevity, we omit the formal proofs of these results and instead provide the necessary intuition, since they are mostly adaptations of the previous proofs.
RR On A Random Marginal (MargRR). In MargRR, user i materializes a random marginal $\beta_i \in C$, then perturbs it using PRR.
Perturbation: User i samples a random marginal $\beta_i \in C$, and evaluates its 2^k indices ($C^\beta(t_i)$) on her input. Note that $C^\beta(t_i)$ is also sparse. The user then perturbs each index of $C^\beta(t_i)$ with $\frac{\epsilon}{2}$-RR (PRR) and sends the tuple $\langle C^\beta(t_i^*), \beta_i \rangle$ 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 $1/p_s = \binom{d}{k} = O(d^k)$. Thus, via Theorem 4.3, we obtain a bound on the error in each entry of each marginal of $\tilde{O}(\sqrt{d^k}/\epsilon\sqrt{N})$. Summing this over the 2^k entries in the marginal, we obtain a total error of $\tilde{O}\left(\frac{2^k d^{k/2}}{\epsilon\sqrt{N}}\right)$.
PS On A Random Marginal (MargPS). As an alternative approach to MargRR, we can use preferential sampling (Section 3.1) 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 behavior 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 $\tilde{O}\left(\frac{2^{3k/2}d^{k/2}}{\epsilon\sqrt{N}}\right)$. 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 behavior. 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 $p_s = O((2d)^{-k})$. We obtain that $\sigma^2 = O((2d)^k/\epsilon^2)$ and $M = O((2d)^k/\epsilon)$. Thus, we bound the absolute error in each reconstructed coefficient by $\tilde{O}\left(\frac{d^{k/2}}{\epsilon\sqrt{N}}\right)$, by invoking Theorem 4.2 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 $\tilde{O}(\frac{2^{3k/2}d^{k/2}}{\epsilon\sqrt{N}})$, 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 4.6. *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 $\tilde{O}(\frac{2^{3k/2}d^{k/2}}{\epsilon\sqrt{N}})$. For MargRR, the bound is $\tilde{O}(\frac{2^k d^{k/2}}{\epsilon\sqrt{N}})$.*

Comparison of all methods. Comparing all six methods, a dependence on a factor of $2^{k/2}\epsilon\sqrt{N}$ 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 2 summarizes these bounds, showing the communication cost (in bits), along with the leading error behavior (supressing logarithmic factors and the common factor of $\epsilon/\sqrt{N}$).

Table 2: Summary of communication and error bounds.

Method	Communication cost	Error behavior
InpRR	2^d	$2^{k/2}2^d$
InpPS	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	$d+k+1$	$2^{3k/2}d^{k/2}$

4.4 Expectation-Maximization Heuristic

While materialization of marginals has not been the primary focus of prior work, a recent paper due to Fanti *et al.* does suggest an alternative approach for the 2-way marginal case [20]. The central idea is for each user to materialize information on all d attributes, and to use post-processing on the observed combinations of reported values to reach an estimate for a given marginal.

In more detail, each user independently perturbs each of the d (binary) attributes via (ϵ/d)-randomized response, i.e. using Budget Sharing (BS). To reconstruct a target marginal distribution, the aggregator applies an instance of Expectation Maximization (EM). 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. As noted by Bassily and Smith [6], this is a plausible heuristic, but does not provide any worst case guarantees of accuracy. 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.[2]. We compare InpEM with best of our methods in Section 5.4.

5 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.

We give our main experimental results on varying parameters N, k and d for our algorithms in this section. Additional experiments on ϵ and alternative primitives are given in Appendix B.

[2]Figure 3 in the Appendix quantifies this in more detail, and shows some parameter settings where the method fails universally.

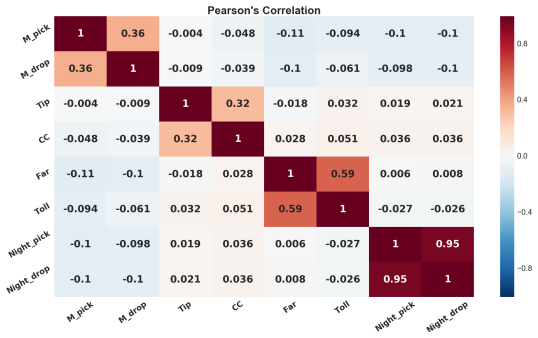

Figure 3: Attribute correlation heatmap of NYC taxi data

5.1 Experimental Setting

Datasets used. We use two sample datasets for our experiments: *NYC Taxi Data [1].* 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 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. Figure 3 gives a heatmap for the strength of pairwise associations using the Pearson coefficient.
Movielens [25]. 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 $t_i \in \{0,1\}^d$. 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 sample (with replacement) a set of random unique records/users ($50K \le N \le 0.5M$) 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. [34] studies a smaller regime of ϵ values, at the expense of a much larger user population. Our experiments adopt the probability settings for Parallel Randomized Response due to Wang *et al.* [41], although we find that these make little difference compared to the "vanilla" Randomized Response probabilities. 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. Per our motivation (Section 1), 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.

5.2 Impact of varying population size N

We aim to understand how much a privately reconstructed marginal $C^\beta(t^*)$ deviates from its non-private counterpart $C^\beta(t)$ when β is drawn from the set of k-way marginals. First, we fix $\epsilon = 1.1$ 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 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 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 RR and PS 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 behavior. 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,

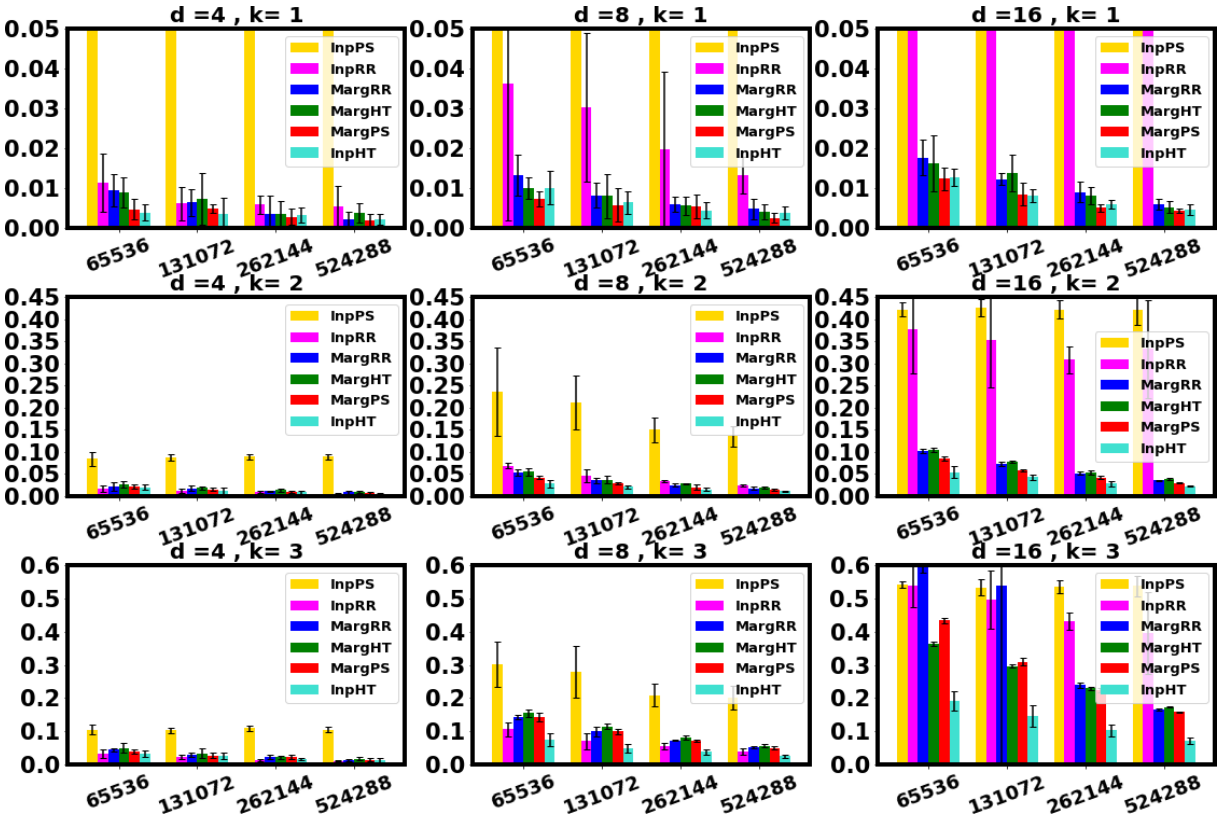

Figure 4: Mean total variation distance for $1, 2, 3$–way marginals over the movielens dataset as N varies

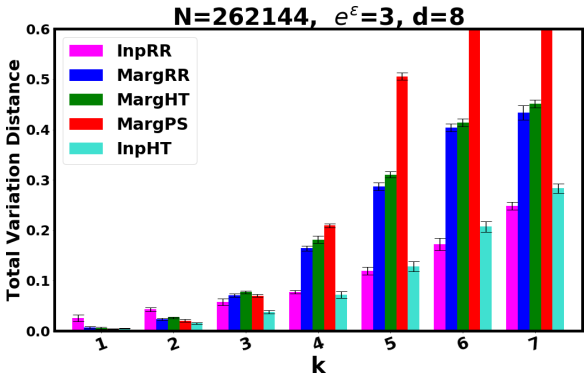

Figure 5: Effect of varying k.

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.

5.3 Impact of increasing marginal size k

In this work, 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 $N = 2^{18}$, $e^\epsilon = 3$, $d = 8$ and vary k from 1 to 7 (Figure 5). 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.

5.4 Impact of increasing dimensionality d

Experimental Setting. Now that we have established the relative performance of our algorithms, we compare to an alternative method that works in the case $k = 2$, denoted InpEM (Section 4.4). We consider a larger range of values of the dimensionality

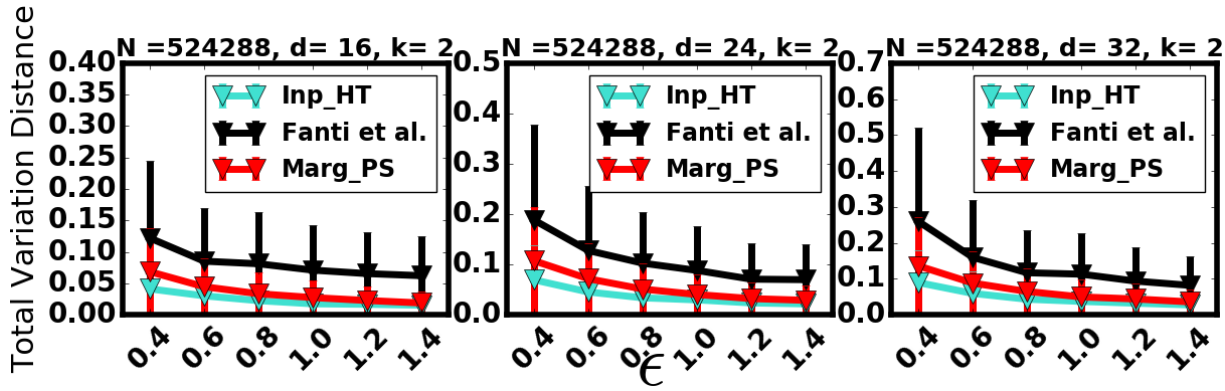

Figure 6: Total variation distance for $k = 2$ on NYC Taxi Trips Data For larger d's.

d, (achieved by duplicating columns) and show the results in Figure 6. 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 convergance criteria are immediately met by the uniform distribution, which is far from the true marginal. We omit formal timing results for brevity; however, convergence time was observed to grow linearly with d. 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. We performed a set of experiments on this approach (details omitted for space reasons); our conclusion is that while this can be applied to our algorithms, there is no improvement compared to the direct construction of unbiased estimators.

6 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.

6.1 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. [3]). It then compares this value to the critical value p for a given confidence 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, the χ^2 statistic is $\sum_{j\in\{0,1\}^2} \frac{(t[j]-\mathbb{E}[t[j]])^2}{\mathbb{E}[t[j]]}$, where $\mathbb{E}[t[j]]$ is the expected value at $t[j]$.

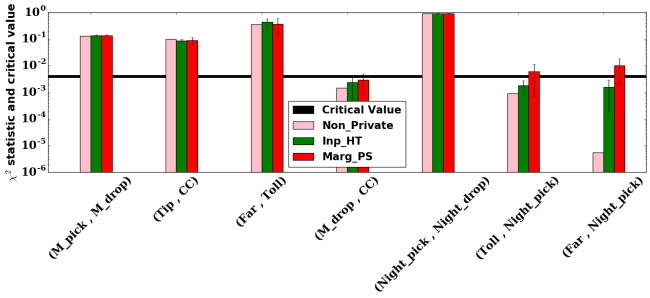

Figure 7: χ^2 test values on $N = 256K$ NYC taxi trips, $\epsilon = 1.1$.

Experimental setting. We use the taxi data for supporting this task since this dataset has a good mix of correlated/weakly correlated attributes (Figure 3). As mentioned above, 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.

Experimental observations. Figure 7 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 [21] 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.

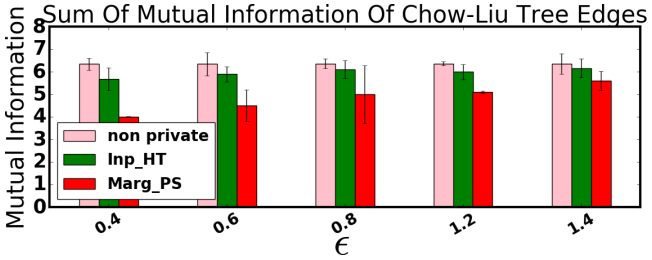

Figure 8: Total mutual information of trees on movielens

6.2 Bayesian Modeling

Exact estimation of a joint distribution for d discrete variables could be computationally infeasible for large d's. Chow and Liu in [9] proposed an algorithm for approximating a joint distribution of a set of discrete variables using products of distributions involving no more than pairs of variables. Since each variable in the approximation depends on at most one more variable, the task of finding such approximation can be thought as finding a tree that optimizes a particular distance metric. They prove that a tree configuration that maximizes 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 center 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

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

Experimental setting. Note that the Chow-Liu algorithm finds a tree from the equivalence class of trees fitting the given data and are not 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 8 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.

6.3 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 two approaches to handling such data.

Binary encoding methods using our algorithms. Many of our algorithms such MargRR, MargPS, InpPS, InpRR 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 6.1. *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,*

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

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 5). 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 5).

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* [17] 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 CONCLUDING REMARKS

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, communciation 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 in browsers and mobile devices: it would require only small modifications to RAPPOR or iOS to incorporate them.

REFERENCES

[1] NYC taxi and limousine commission, trip record data, 2017.

[2] A Thakurta, A Vyrros, U Vaishampayan, G Kapoor, J Freudiger, V Rangarajan Sridhar, D Davidson. Private dictionary population satisfying local differential privacy, March 2017. US Patent 9,594,741 B1.

[3] A. Agresti. *Categorical Data Analysis*. Wiley Series in Probability and Statistics. Wiley, 2003.

[4] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. 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.

[5] R. Bassily, K. Nissim, U. Stemmer, and A. G. Thakurta. Practical locally private heavy hitters. In *NIPS*, pages 2285–2293, 2017.

[6] R. Bassily and A. Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 127–135. ACM, 2015.

[7] A. Chaudhuri and R. Mukerjee. *Randomized response: Theory and techniques*. Marcel Dekker, 1988.

[8] R. Chen, H. Li, A. K. Qin, S. P. Kasiviswanathan, and H. 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.

[9] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, May 1968.

[10] Differential Privacy Team, Apple. Learning with privacy at scale. 2017.

[11] B. Ding, J. Kulkarni, and S. Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems 30*, December 2017.

[12] B. Ding, M. Winslett, J. Han, and Z. 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.

[13] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *PODS*. ACM, 2003.

[14] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. In *FOCS*. IEEE, 2013.

[15] C. Dwork, F. McSherry, K. Nissim, and A. 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.

[16] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Annual International Cryptology Conference*. Springer, 2004.

[17] B. Efron and C. Stein. The jackknife estimate of variance. *The Annals of Statistics*, pages 586–596, 1981.

[18] U. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *ACM CCS*, 2014.

[19] A. V. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *ACM SIGKDD*, pages 217–228, 2002.

[20] G. Fanti, V. Pihur, and Ú. 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.

[21] M. Gaboardi, R. M. Rogers, and S. P. Vadhan. Differentially private chi-squared hypothesis testing: Goodness of fit and independence testing. 2016.

[22] A. Gupta, M. Hardt, A. Roth, and J. Ullman. Privately releasing conjunctions and the statistical query barrier. *SIAM Journal on Computing*, 42(4):1494–1520, 2013.

[23] M. Hardt, K. Ligett, and F. 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.

[24] M. Hardt and G. 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.

[25] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, Dec. 2015.

[26] D. Jurafsky and J. Martin. *Speech and Language Processing*. Always learning. Pearson, 2014.

[27] P. Kairouz, S. Oh, and P. Viswanath. Differentially private multi-party computation: Optimality of non-interactive randomized response. *arXiv preprint arXiv:1407.1546*, 2014.

[28] P. Kairouz, S. Oh, and P. Viswanath. Extremal mechanisms for local differential privacy. In *Advances in neural information processing systems*, pages 2879–2887, 2014.

[29] T. K. Leen, T. G. Dietterich, and V. Tresp, editors. *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*. MIT Press, 2001.

[30] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE International Conference on Data Engineering*, 2007.

[31] A. Machanavajjhala, X. He, and M. Hay. Differential privacy in the wild: A tutorial on current practices & open challenges. *Proceedings of the VLDB Endowment*, 9(13):1611–1614, 2016.

[32] F. McSherry and K. 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.

[33] A. Narayan. *Distributed differential privacy and applications*. University of Pennsylvania, 2015.

[34] T. T. Nguyên, X. Xiao, Y. Yang, S. C. Hui, H. Shin, and J. Shin. Collecting and analyzing data from smart device users with local differential privacy. *CoRR*, abs/1606.05053, 2016.

[35] R. O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, New York, NY, USA, 2014.

[36] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. 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.

[37] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, SRI, 1998.

[38] A. D. Sarwate and K. Chaudhuri. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE Signal Process. Mag.*, 30(5):86–94, 2013.

[39] J. Thaler, J. Ullman, and S. Vadhan. Faster algorithms for privately releasing marginals. In *International Colloquium on Automata, Languages, and Programming*, pages 810–821. Springer, 2012.

[40] S. Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer, 2017.

[41] T. Wang, J. Blocki, N. Li, and S. 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.

[42] S. L. Warner. Randomised response: a survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, Mar. 1965.

[43] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. In *ICDE*, 2010.

A DEFERRED PROOFS

In this appendix, we provide the detailed technical proofs for the claimed privacy and accuracy properties of our algorithms.

Proof of Theorem 4.2. 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 $t_i[j] \in \{-1, 1\}$ be i's unknown true input at location j and $t_i^*[j]$ be the unbiased estimate of $t_i[j]$. First, we derive the values we should ascribe to t^* to ensure unbiasedness, i.e. $\mathsf{E}[t_i^*[j]] = t_i[j]$.

1. When j is sampled (with probability p_s) and $t_i[j] = 1$, we set $t_i^*[j] = x/p_s$ with probability p_r and $t_i^*[j] = y/p_s$ otherwise.
2. When j is sampled (with probability p_s), and $t_i[j] = -1$, we set $t_i^*[j] = y/p_s$ with probability p_r and x/p_s otherwise.
3. When j is not sampled, we implicitly set $t_i^*[j] = 0$.

We can encode these conditions with linear equations:

$$p_r x + (1 - p_r)y = -1 \tag{5}$$

$$p_r y + (1 - p_r)x = 1 \tag{6}$$

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 $Y_i[j] = t_i^*[j] - t_i[j]$. Observe that $\mathsf{E}[Y_i[j]]$ 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 $t_i[j] = 1$ or -1. Then:

$$\begin{aligned}
\mathsf{Var}[Y_i[j]] &= \mathsf{E}[Y_i^2[j]] \\
&= \frac{p_r p_s}{p_s^2}\Big|\frac{1}{2p_r - 1} - 1\Big|^2 + \frac{(1-p_r)p_s}{p_s^2}\Big|1 + \frac{1}{2p_r - 1}\Big|^2 + (1-p_s)1^2 \\
&\le \frac{p_r}{p_s}\left(\frac{2p_r - 2}{2p_r - 1}\right)^2 + \frac{(1-p_r)}{p_s}\left(\frac{2p_r}{2p_r - 1}\right)^2 + (1-p_s) \\
&= \frac{4}{p_s(2p_r-1)^2}(p_r(1-p_r)^2 + (1-p_r)p_r^2) + (1-p_s) \\
&= \frac{4p_r(1-p_r)}{p_s(2p_r-1)^2} + (1-p_s) := \sigma^2. \qquad (7)
\end{aligned}$$

Now we consider the effect of aggregating N estimates of the j'th population parameter. Using Bernstein's inequality (Definition 4.1), 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.

$$\begin{aligned}
\Pr\left[\frac{|\sum_{i=1}^N Y_i[j]|}{N} \ge c\right] &\le 2\exp\left(-\frac{Nc^2}{2\sigma^2 + \frac{2cM}{3}}\right) \\
&\le 2\exp\left(-\frac{Nc^2}{2(\frac{p_r(1-p_r)}{p_s(2p_r-1)^2}+1) + \frac{2cp_r}{3p_s(2p_r-1)}}\right) \\
&= 2\exp\left(-\frac{Nc^2}{\frac{2p_r}{p_s(2p_r-1)}(\frac{2(1-p_r)}{(2p_r-1)} + \frac{c}{3}) + 2}\right) \qquad (8)
\end{aligned}$$

This provides us with the statement of the theorem. □

Proof of Theorem 4.3. 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 $p_r = \frac{e^{\epsilon/2}}{1+e^{\epsilon/2}}$, and $p_s = 1$. For the purpose of analysis only, we reduce the problem so that we can apply Theorem 4.2, by applying a remapping from $\{0,1\}$ to $\{-1,1\}$: we replace $t_i[j]$ with $t_i'[j] = 2t_i[j] - 1$. Observe that the absolute error in reconstructing $t_i'[j]$ is only a constant factor of that in reconstructing $t_i'[j]$. Writing $\alpha = e^{\epsilon/2}$, then we have the variance of the local errors $Y_i[j] = (t_i[j] - t_i^*[j])$ is (substituting these values of p_r and p_s into (7)):

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

The reconstruction of the full input distribution is $t^* = \sum_{i=1}^N t_i^*/N$. We can make use of the inequalities $\frac{1}{e^{\epsilon/2}-1} \le \frac{1}{\epsilon}$ and $1 < e^{\epsilon/2} < 4$ for $0 < \epsilon < 2$ to bound the variance and substitute into (8).

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

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

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

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 $\tilde{O}(\frac{1}{\epsilon}\sqrt{\frac{1}{N}})$, by applying a union bound.

We construct the target marginal β via the marginal operator, so $\widehat{C^\beta} = C^\beta(t^*)$. Each entry $t^*[j]$ is an unbiased estimator for $t[j]$ whose absolute value is bounded by c with probability $1-\delta$. Conditioning on this event, we compute $\widehat{C^\beta}[\gamma] = \sum_{\alpha \preceq \gamma} t^*[\alpha]$, summing over the 2^{d-k} values of $\alpha \preceq \gamma$. The error in this quantity is then at most $\tilde{O}(c\sqrt{2^{d-k}})$, applying a Hoeffding bound (Definition 4.1). 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 $\tilde{O}(\frac{2^k 2^{(d-k)/2}}{\epsilon\sqrt{N}}) = \tilde{O}(\frac{2^{(d+k)/2}}{\epsilon\sqrt{N}})$. □

Proof of Theorem 4.4. Similar to Theorem 4.2, 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 PS, 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 4.1, 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) $t_i[j] = 1$. 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 $|t_i^*[j] = t_i[j]|$.
(ii) $t_i[j] = 0$. 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 $t_i[j]$. Note that an upper bound M on $Y_i[j]$ is D/Δ. We compute bounds on Y_i^2, conditioned on $t_i[j]$.

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

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

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

Using this in the Bernstein inequality (Definition 4.1), we obtain

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

If we write $\Psi_j = \sqrt{f_jD + 1 - f_j}$, 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 C^β to the reconstructed input t^*, each of the 2^k entries is formed by summing up $(D+1)/2^k$ (unbiased) entries of t^*. 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 4.1), we obtain that each $C^\beta(t^*)[\gamma]$ 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,

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

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 $p_s = (1+De^{-\epsilon})^{-1}$. Then

$$\frac{D}{\Delta} = \frac{D}{(D+1)/(1+De^{-\epsilon}) - 1} = \frac{D(1+De^{-\epsilon})}{D+1-1-De^{-\epsilon}}$$
$$= \frac{1+De^{-\epsilon}}{1-e^{-\epsilon}} = \frac{1}{1-e^{-\epsilon}} + \frac{D}{e^\epsilon - 1}$$

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

Proof of Theorem 4.5. The proof proceeds along the same lines as for Theorem 4.3. We set $p_r = e^\epsilon/(1+e^\epsilon)$ to ensure that InpHT meets ϵ-LDP. Recall that, from Lemma 3.6, our aim is to compute Hadamard coefficients as the normalized sum of the coefficients from each user. To apply the Master theorem (Theorem 4.2), we first multiply up each coefficient by the $2^{d/2}$ factor from the Hadamard coefficients θ (Definition 3.5). 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

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

Substituting this variance bound into (8), 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 3.7, that the marginal can be expressed in terms of a linear sum of Hadamard coefficients. Adapting (4), we have that

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

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) $2^k2^{k/2-d/2}c2^{d/2-k} = c2^{k/2} = \tilde{O}\left(\frac{2^{k/2}\sqrt{T}}{\epsilon\sqrt{N}}\right)$. □

Table 3: Failure rate for InpEM on taxi dataset for small ϵ

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

B ADDITIONAL EXPERIMENTS

In this section, we provide additional experimental insight and comparison. Table 3 shows experiments using the method of Fanti *et al.* [20]. We then consider the impact of varying ϵ, and some additional comparisons with "frequency oracles".

B.1 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 9.

Observations: We observe a decline in error as we increase the privacy budget ϵ. Once again we see that InpPS, InpRR, MargRR

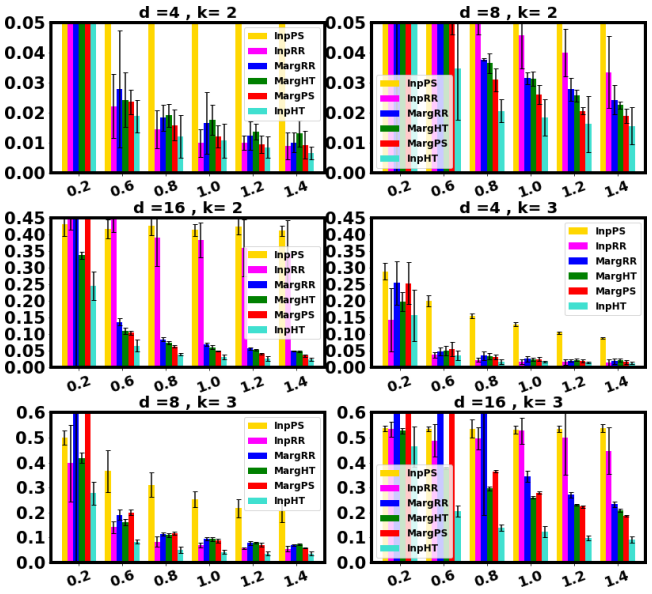

Figure 9: Mean total variation for $1, 2, 3$–way marginals for $N = 256K$ movielens users as ϵ varies.

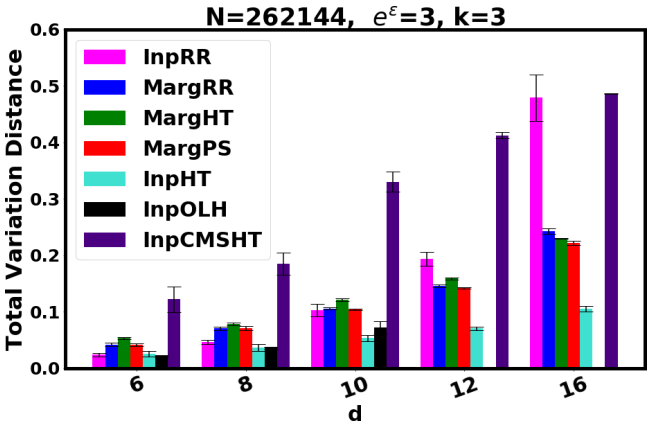

Figure 10: Effect of varying d with frequency oracles

are unfavorable 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.

B.2 Frequency Oracle Methods

As discussed in Section 2, there have been several recent papers addressing the problem of estimating population frequencies under LDP [2, 5, 41]. These works provide a "frequency oracle": an LDP protocol which allows the frequency of any element from a large domain to be estimated accurately. 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. 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) [41]: The OLH primitive satisfying ϵ–LDP proposed by Wang et al. handles large domain size via universal hash functions. In summary, each user $i \in [N]$ with a sparse input $t_i \in \mathcal{I}_{2^m \times 2^m}$ uniformly samples a hash function h_i from a family $\mathcal{H} : [2^m] \to [1 + e^\epsilon]$ 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 PS. 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 $O(\epsilon)$ bits, but the aggregator's time cost is $O(2^d)$ per user.

Private Hadamard Count-Min Sketch (InpHTCMS) [10]: The method deployed by Apple adapts ideas from sketching, and is also similar to a related method [5]. 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 $t_i \in \mathcal{I}_{2^m \times 2^m}$ 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 10). 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.