

# Faster Private Release of Marginals on Small Databases

Karthekeyan Chandrasekaran<sup>\*</sup>  
School of Engineering and  
Applied Sciences  
Harvard University  
karthe@seas.harvard.edu

Justin Thaler<sup>†</sup>  
School of Engineering and  
Applied Sciences  
Harvard University  
jthaler@seas.harvard.edu

Jonathan Ullman<sup>‡</sup>  
School of Engineering and  
Applied Sciences  
Harvard University  
jullman@seas.harvard.edu

Andrew Wan<sup>§</sup>  
Simons institute for the Theory  
of Computing  
U.C. Berkeley  
atw12@seas.harvard.edu

## ABSTRACT

We study the problem of answering  $k$ -way marginal queries on a database  $D \in (\{0, 1\}^d)^n$ , while preserving differential privacy. The answer to a  $k$ -way marginal query is the fraction of the database's records  $x \in \{0, 1\}^d$  with a given value in each of a given set of up to  $k$  columns. Marginal queries enable a rich class of statistical analyses on a dataset, and designing efficient algorithms for privately answering marginal queries has been identified as an important open problem in private data analysis. For any  $k$ , we give a differentially private online algorithm that runs in time  $\text{poly}(n, 2^{o(d)})$  per query and answers any sequence of  $\text{poly}(n)$  many  $k$ -way marginal queries with error at most  $\pm 0.01$  on every query, provided  $n \gtrsim d^{0.51}$ . To the best of our knowledge, this is the first algorithm capable of privately answering marginal queries with a non-trivial worst-case accuracy guarantee for databases containing  $\text{poly}(d, k)$  records in time  $\exp(o(d))$ . Our algorithm runs the private multiplicative weights algorithm (Hardt and Rothblum, FOCS '10) on a new approximate polynomial representation of the database.

We derive our representation for the database by approximating the OR function restricted to low Hamming weight

<sup>\*</sup>Supported by Simons Fellowship.

<sup>†</sup>Supported by an NSF Graduate Research Fellowship and NSF grants CNS-1011840 and CCF-0915922.

<sup>‡</sup>Supported by NSF grant CNS-1237235 and a Siebel Scholarship.

<sup>§</sup>On leave from IIS, Tsinghua University. This work was completed while at Harvard University and supported by NSF grant CCF-0964401 and NSFC grant 61250110218.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
ITCS'14, January 12–14, 2014, Princeton, New Jersey, USA.  
Copyright 2014 ACM 978-1-4503-2698-8/14/01 ...\$15.00.  
<http://dx.doi.org/10.1145/2554797.2554833>.

inputs using low-degree polynomials with coefficients of bounded  $L_1$ -norm. In doing so, we show new upper and lower bounds on the degree of such polynomials, which may be of independent approximation-theoretic interest.

## 1. INTRODUCTION

Consider a database  $D \in (\{0, 1\}^d)^n$  in which each of the  $n (= |D|)$  rows corresponds to an individual's record, and each record consists of  $d$  binary attributes. The goal of privacy-preserving data analysis is to enable rich statistical analyses on the database while protecting the privacy of the individuals. In this work, we seek to achieve *differential privacy* [10], which guarantees that no individual's data has a significant influence on the information released about the database.

One of the most important classes of statistics on a dataset is its *marginals*. A marginal query is specified by a set  $S \subseteq [d]$  and a pattern  $t \in \{0, 1\}^{|S|}$ . The query asks, "What fraction of the individual records in  $D$  has each of the attributes  $j \in S$  set to  $t_j$ ?" A major open problem in privacy-preserving data analysis is to *efficiently* release a differentially private summary of the database that enables analysts to answer each of the  $3^d$  marginal queries. A natural subclass of marginals are  $k$ -way marginals, the subset of marginals specified by sets  $S \subseteq [d]$  such that  $|S| \leq k$ .

Privately answering marginal queries is a special case of the more general problem of privately answering *counting queries* on the database, which are queries of the form, "What fraction of individual records in  $D$  satisfy some property  $q$ ?" Early work in differential privacy [2, 9, 10] showed how to privately answer any set of counting queries  $\mathcal{Q}$  approximately by perturbing the answers with appropriately calibrated noise, ensuring good accuracy (say, within  $\pm 0.01$  of the true answer) provided  $|D| \gtrsim |\mathcal{Q}|^{1/2}$ .

However, in many settings data is difficult or expensive to obtain, and the requirement that  $|D| \gtrsim |\mathcal{Q}|^{1/2}$  is too restrictive. For instance, if the query set  $\mathcal{Q}$  includes all  $k$ -way marginal queries then  $|\mathcal{Q}| \geq d^{\Theta(k)}$ , and it may be impractical to collect enough data to ensure  $|D| \gtrsim |\mathcal{Q}|^{1/2}$ , even for moderate values of  $k$ . Fortunately, a remarkable line of work initiated by Blum et al. [3] and continuing

with [11, 14, 17–19, 21, 28], has shown how to privately release approximate answers to any set of counting queries, even when  $|\mathcal{Q}|$  is exponentially larger than  $|D|$ . For example, the *online private multiplicative weights algorithm* of Hardt and Rothblum [19] gives accurate answers to any (possibly adaptively chosen) sequence of queries  $\mathcal{Q}$  provided  $|D| \gtrsim \sqrt{d} \log |\mathcal{Q}|$ . Hence, if the sequence consists of all  $k$ -way marginal queries, then the algorithm will give accurate answers provided  $|D| \gtrsim k\sqrt{d}$ . Unfortunately, all of these algorithms have running time at least  $2^d$  per query, even in the simplest setting where  $\mathcal{Q}$  is the set of 2-way marginals.

Given this state of affairs, it is natural to seek *efficient* algorithms capable of privately releasing approximate answers to marginal queries even when  $|D| \ll d^k$ . The most efficient algorithm known for this problem, due to Thaler, Ullman, and Vadhan [34] (building on the work of Hardt, Rothblum, and Servedio [20]) runs in time  $d^{O(\sqrt{k})}$  and releases a summary from which an analyst can compute the answer to any  $k$ -way marginal query in time  $d^{O(\sqrt{k})}$ .

Even though  $|D|$  can be much smaller than  $|\mathcal{Q}|^{1/2}$ , a major drawback of this algorithm and other efficient algorithms for releasing marginals (e.g. [7, 12, 15, 16, 20]) is that the database still must be significantly larger than  $\tilde{\Theta}(k\sqrt{d})$ , which we know would suffice for inefficient algorithms. Recent experimental work of Hardt, Ligett, and McSherry [18] demonstrates that for some databases of interest, even the  $2^d$ -time private multiplicative weights algorithm is practical, and also shows that more efficient algorithms based on adding independent noise do not provide good accuracy for these databases. Motivated by these findings, we believe that an important approach to designing practical algorithms is to achieve a minimum database size comparable to that of private multiplicative weights, and seek to optimize the running time of the algorithm as much as possible. In this paper we give the first algorithms for privately answering marginal queries for this parameter regime.

## 1.1 Our Results

In this paper we give faster algorithms for privately answering marginal queries on databases of size  $\tilde{O}(d^{0.51}/\varepsilon)$ , which is nearly the smallest a database can be while admitting any differentially private approximation to marginal queries [6].

**THEOREM 1.1.** *There exists a constant  $C > 0$  such that for every  $k, d, n \in \mathbb{N}$ ,  $k \leq d$ , and every  $\varepsilon, \delta > 0$ , there is an  $(\varepsilon, \delta)$ -differentially private online algorithm that, on input a database  $D \in (\{0, 1\}^d)^n$ , runs in time*

$$\text{poly} \left( n, \min \left\{ \exp \left( d^{1-1/C\sqrt{k}} \right), \exp \left( d / \log^{0.99} d \right) \right\} \right)$$

*per query and answers any sequence  $\mathcal{Q}$  of (possibly adaptively chosen)  $k$ -way marginal queries on  $D$  up to an additive error of at most  $\pm 0.01$  on every query with probability at least 0.99, provided that  $n \geq Cd^{0.51} \log |\mathcal{Q}| \log(1/\delta)/\varepsilon$ .*

When  $k$  is much smaller than  $d$ , it may be useful to view our algorithm as an offline algorithm for releasing answers to all  $k$ -way marginal queries. This offline algorithm can be obtained simply by requesting answers to each of the  $d^{\Theta(k)}$  distinct  $k$ -way marginal queries from the online mechanism. In this case we obtain the following corollary.

**COROLLARY 1.2.** *There exists a constant  $C > 0$  such that for every  $k, d, n \in \mathbb{N}$ ,  $k = O(d/\log d)$ , and every  $\varepsilon, \delta > 0$ , there is an  $(\varepsilon, \delta)$ -differentially private offline algorithm that, on input a database  $D \in (\{0, 1\}^d)^n$ , runs in time*

$$\text{poly} \left( n, \min \left\{ \exp \left( d^{1-1/C\sqrt{k}} \right), \exp \left( d / \log^{0.99} d \right) \right\} \right)$$

*and, with probability at least 0.99, releases answers to every  $k$ -way marginal query on  $D$  up to an additive error of at most  $\pm 0.01$ , provided that  $n \geq Ckd^{0.51} \log(1/\delta)/\varepsilon$ .*

Here  $\binom{d}{\leq k} := \sum_{i=0}^k \binom{d}{i}$ , and the number of  $k$ -way marginals on  $\{0, 1\}^d$  is bounded by a polynomial in this quantity. See Table 1 for a comparison of relevant results on privately answering marginal queries.

### Remarks.

1. When  $k = \Omega(\log^2 d)$ , the minimum database size requirement can be improved to

$$n \geq Ckd^{0.5+o(1)} \log(1/\delta)/\varepsilon,$$

but we have stated the theorems with a weaker bound for simplicity. (Here  $C > 0$  is a universal constant and the  $o(1)$  is with respect to  $d$ .)

2. Our algorithm can be modified so that instead of releasing approximate answers to each  $k$ -way marginal explicitly, it releases a summary of the database of size  $\tilde{O}(kd^{0.01})$  from which an analyst can compute an approximate answer to any  $k$ -way marginal in time  $\tilde{O}(kd^{1.01})$ .

A key ingredient in our algorithm is a new approximate representation of the database using polynomial approximations to the  $d$ -variate OR function restricted to inputs of Hamming weight at most  $k$ . For any such polynomial, the degree determines the runtime of our algorithm, while the  $L_1$ -weight of the coefficient vector determines the minimum required database size. Although low-degree low  $L_1$ -weight polynomial approximations to the OR function have been studied in the context of approximation theory and learning theory [29], our setting requires an approximation only over a restricted subset of the inputs. When the polynomial needs to approximate the OR function only on a subset of the inputs, is it possible to reduce the degree and  $L_1$ -weight (in comparison to [29]) of the polynomial?

Our main technical contribution addresses this variant of the polynomial approximation problem. We believe that our construction of such polynomials (Theorem 1.3) as well as the lower bound (Theorem 1.4) could be of independent approximation-theoretic interest. The following theorem shows a construction of polynomials that achieve better degree and  $L_1$ -weight in comparison to [29] for small values of  $k$ . Let  $\text{OR}_d : \{-1, 1\}^d \rightarrow \{-1, 1\}$  denote the OR function on  $d$  variables with the convention that  $-1$  is TRUE, and for any vector  $x \in \{-1, 1\}^d$ , let  $|x|$  denote the number of coordinates of  $x$  equal to  $-1$ .

**THEOREM 1.3.** *Let  $k \in [d]$ . For some constant  $C > 0$ , there exists a polynomial  $p$  such that*

$$(i) \quad |p(x) - \text{OR}_d(x)| \leq 1/400 \text{ for every } x \in \{-1, 1\}^d : |x| \leq k,$$

Reference	Running Time per Query	Database Size
[2, 9, 10, 13]	$O(1)$	$\tilde{O}(d^{k/2})$
[17, 19]	$2^{O(d)}$	$\tilde{O}(k\sqrt{d})$
[20, 34]	$d^{O(\sqrt{k})}$	$d^{O(\sqrt{k})}$
[12]	$d^{O(k)}$	$\tilde{O}(d^{\lceil k/2 \rceil / 2})$
This paper	$2^{O(d/\log^{0.99} d)}$	$kd^{0.5+o(1)}$
This paper	$2^{d^{1-\Omega(1/\sqrt{k})}}$	$\tilde{O}(kd^{.51})$

**Table 1: Prior results on differentially private release of  $k$ -way marginals with error  $\pm 0.01$  on every marginal. (The running time ignores dependence on the database size, privacy parameters, and the time required to evaluate the query non-privately.)**

(ii) the  $L_1$ -weight of the coefficient vector of  $p$  is at most  $d^{0.01}$ , and

(iii) the degree of  $p$  is at most

$$\min \left\{ d^{1-\frac{1}{c\sqrt{k}}}, \frac{d}{\log^{0.995} d} \right\}.$$

The degree bound of  $d/\log^{0.995} d$  in the above theorem follows directly from techniques developed in [29], while the degree bound of  $d^{1-\frac{1}{c\sqrt{k}}}$  requires additional insight. We also show a lower bound to exhibit the tightness of our construction.

**THEOREM 1.4.** *Let  $k = o(\log d)$ , and let  $p$  be a real  $d$ -variate polynomial satisfying  $|p(x) - OR_d(x)| \leq 1/6$  for all  $x \in \{-1, 1\}^d$  with  $|x| \leq k$ . If the  $L_1$ -weight of the coefficient vector of  $p$  is  $d^{O(1)}$ , then the degree of  $p$  is at least  $d^{1-O(1/\sqrt{k})}$ .*

We note that our algorithmic approach for designing efficient private data release algorithms would work equally well if we have any small set of functions whose low- $L_1$ -weight linear combinations approximate disjunctions restricted to inputs of Hamming weight at most  $k$ . Our lower bound limits the applicability of our approach if we choose to use low-degree monomials as the set of functions. We observe that this also rules out several natural candidates that can themselves be computed exactly by a low-weight polynomial of low-degree (e.g., the set of small-width conjunctions). There is some additional evidence from prior work that low-degree monomials may be the optimal choice: if we only care about the *size* of the set of functions used to approximate disjunctions on inputs of Hamming weight at most  $k$ , then prior work shows that low-degree monomials are indeed optimal [31] (see also Section 5 in the full version of [34]). It remains an interesting open question to determine whether this optimality still holds when we restrict the  $L_1$  weight of the linear combinations used in the approximations to be  $\text{poly}(d)$ .

## 1.2 Techniques

For notational convenience, we focus on *monotone  $k$ -way disjunction queries*. However, our results extend straightforwardly to general non-monotone  $k$ -way marginal queries via simple transformations on the database and queries. A monotone  $k$ -way disjunction is specified by a set  $S \subseteq [d]$  of size  $k$  and asks what fraction of records in  $D$  have at least one of the attributes in  $S$  set to 1.

Following the approach introduced by Gupta et al. [16] and developed into a general theory in [20], we view the problem of releasing answers to conjunction queries as a learning problem. That is, we view the database as specifying a function  $f_D: \{-1, 1\}^d \rightarrow [0, 1]$ , in which each input vector  $s \in \{-1, 1\}^d$  is interpreted as the indicator vector of a set  $S \subseteq \{1, \dots, d\}$ , with  $s_i = -1$  iff  $i \in S$ , and  $f_D(s)$  equals the evaluation of the conjunction query specified by  $S$  on the database  $D$ . Then, our goal is to privately learn to approximate the function  $f_D$ ; this is accomplished in [20] by approximating  $f_D$  succinctly with polynomials and learning the polynomial privately. Polynomial approximation is central to our approach as well, as we explain below.

We begin with a description of how the parameters of the online learning algorithm determine the parameters of the online differentially private learning algorithm. We consider the “IDC framework” [17]—which captures the private multiplicative weights algorithm [19] among others [17, 21, 28]—for deriving differentially private online algorithms from any *online learning algorithm* that may not necessarily be privacy preserving.

Informally, an online learning algorithm is one that takes a (possibly adaptively chosen) sequence of inputs  $s_1, s_2, \dots$  and returns answers  $a_1, a_2, \dots$  to each, representing “guesses” about the values  $f_D(s_1), f_D(s_2), \dots$  for the unknown function  $f_D$ . After making each guess  $a_i$ , the learner is given some information about the value of  $f_D(s_i)$ . The quantities of interest are the running time required by the online learner to produce each guess  $a_i$  and the number of “mistakes” made by the learner, which is the number of rounds  $i$  in which  $a_i$  is “far” from  $f_D(s_i)$ . Ultimately, for the differentially private algorithm derived in the IDC framework, the notion of far will correspond to the accuracy, the per query running time will essentially be equal to the running time of the online learning algorithm, and the minimum database size required by the private algorithm will be proportional to the square root of the number of mistakes.

We next describe the well-known technique of deriving faster online learning algorithms that commit fewer mistakes using *polynomial approximations* to the target function. Indeed, it is well-known that if  $f_D$  can be approximated to high accuracy by a  $d$ -variate polynomial  $p_D: \{-1, 1\}^d \rightarrow \mathbb{R}$  of degree  $t$  and  $L_1$ -weight at most  $W$ , where the weight is defined to be the sum of the absolute values of the coefficients, then there is an online learning algorithm that runs in time  $\text{poly}\left(\binom{d}{\leq t}\right)$  and makes  $O(W^2 d)$  mistakes. Thus, if  $t \ll d$ , the running time of such an online learning algorithm will be significantly less than  $2^d$  and the number of mistakes

(and thus the minimum database size of the resulting private algorithm) will only blow up by a factor of  $W$ .

Consequently, our goal boils down to constructing the best possible polynomial representation  $p_D$  for any database  $D$  – one with low-degree, low- $L_1$ -weight such that  $|p_D(s) - f_D(s)|$  is small for all vectors  $s \in \{-1, 1\}^d$  corresponding to monotone  $k$ -way disjunction queries. To accomplish this goal, it is sufficient to construct a low-degree, low- $L_1$ -weight polynomial that can approximate the  $d$ -variate OR function on inputs of Hamming weight at most  $k$  (i.e., those that have  $-1$  in at most  $k$  indices). Such problems are well-studied in the approximation-theory literature, however our variant requires polynomials to be accurate only on a restricted subset of inputs. In fact, the existence of a polynomial with degree  $d/\log^{0.99} d$  and  $L_1$ -weight  $d^{o(1)}$  that approximates the  $d$ -variate OR function on all inputs follows from the work of Servedio et al. [29]. We improve these bounds for small values of  $k$  by constructing an approximating polynomial that has degree  $d^{1-\Omega(1/\sqrt{k})}$  and  $L_1$ -weight  $d^{0.01}$ .

We also prove a new approximation-theoretic lower bound for polynomials that seek to approximate a target function for a restricted subset of inputs. Specifically, we show that for any  $k = o(\log d)$ , any polynomial  $p$  of weight  $\text{poly}(d)$  that satisfies  $|p(s) - \text{OR}(s)| \leq 1/6$  for all inputs  $s \in \{-1, 1\}^d$  of Hamming weight at most  $k$  must have degree  $d^{1-O(1/\sqrt{k})}$ . We prove our lower bound by expressing the problem of constructing such a low-weight, low-degree polynomial  $p$  as a linear program, and exhibiting an explicit solution to the dual of this linear program. Our proof is inspired by recent work of Sherstov [30, 31, 33] and Bun-Thaler [5].

### 1.3 Related Work

**Other Results on Privately Releasing Marginals.** In work subsequent to our result, Dwork et al. [12] show how to privately release marginals in a very different parameter regime. Their algorithm is faster than ours, running in time  $\text{poly}(\binom{d}{\leq k})$ , and has better dependence on the error parameter. However, their algorithm requires that the database size is  $\tilde{\Omega}(d^{\lceil k/2 \rceil / 2})$  for answering with error  $\pm 0.01$ . This size is comparable to the optimal  $\tilde{\Omega}(k\sqrt{d})$  only when  $k \leq 2$ . In contrast, our algorithm has nearly-optimal minimum database size for every choice of  $k$ .

While we have focused on accurately answering *every*  $k$ -way marginal query, or more generally every query in a sequence of marginal queries, several other works have considered more relaxed notions of accuracy. These works show how to efficiently release a summary of the database from which an analyst can efficiently compute an approximate answer to marginal queries, with the guarantee that the *average* error of a marginal query is at most .01, when the query is chosen from a particular distribution. In particular, Feldman and Kothari [15] achieve small average error over the uniform distribution with running time and database size  $\tilde{O}(d^2)$ ; Gupta et al. [16] achieve small average error over any product distribution with running time and minimum database size  $\text{poly}(d)$ ; finally Hardt et al. [20] show how to achieve small average error over arbitrary distributions with running time and minimum database size  $2^{\tilde{O}(d^{1/3})}$ . All of these results are based on the approach of learning the function  $f_D$ .

Several works have also considered information theoretic bounds on the minimum database size required to answer

$k$ -way marginals. Kasiviswanathan et al. [22] showed that  $|D| \geq \min\{1/\alpha^2, d^{k/2}/\alpha\}$  is necessary to answer all  $k$ -way marginals with error  $\pm\alpha$ . De [8] extended this result to hold even when accuracy  $\pm\alpha$  can be violated for a constant fraction of  $k$ -way marginals. In our regime, where  $\alpha = \Omega(1)$ , their results do not give a non-trivial lower bound. In forthcoming work, Bun, Ullman, and Vadhan [6] have proven a lower bound of  $|D| \geq \tilde{\Omega}(k\sqrt{d})$ , which is nearly optimal for  $\alpha = \Omega(1)$ .

**Hardness of Differential Privacy.** Ullman [35] (building on the results of Dwork et al. [11]), showed that any  $2^{o(d)}$ -time differentially private algorithm that answers *arbitrary* counting queries can only give accurate answers if  $|D| \gtrsim |\mathcal{Q}|^{1/2}$ , assuming the existence of exponentially hard one-way functions. Our algorithms have running time  $2^{o(d)}$  and are accurate when  $|D| \ll |\mathcal{Q}|^{1/2}$ , and thus show a separation between answering marginal queries and answering arbitrary counting queries.

When viewed as an offline algorithm for answering all  $k$ -way marginals, our algorithm will return a list of values containing answers to each  $k$ -way marginal query. It would in some cases be more attractive if we could return a *synthetic database*, which is a new database  $\hat{D} \in (\{0, 1\}^d)^n$  whose rows are “fake”, but such that  $\hat{D}$  approximately preserves many of the statistical properties of the database  $D$  (e.g., all the marginals). Some of the previous work on counting query release has provided synthetic data [1, 3, 11, 14, 18].

Unfortunately, Ullman and Vadhan [36] (building on [11]) have shown that no differentially private sanitizer with running time  $\text{poly}(d)$  can take a database  $D \in (\{0, 1\}^d)^n$  and output a private synthetic database  $\hat{D}$ , all of whose 2-way marginals are approximately equal to those of  $D$ , assuming the existence of one-way functions. They also showed that under certain strong cryptographic assumptions, there is no differentially private sanitizer with running time  $2^{d^{1-\Omega(1)}}$  can output a private synthetic database, all of whose 2-way marginals are approximately equal to those of  $D$ . Our algorithms indeed achieve this running time and accuracy guarantee when releasing  $k$ -way marginals for constant  $k$ , and thus it may be inherent that our algorithms do not generate synthetic data.

**Approximation Theory.** Servedio et al. [29] focused on developing low-weight, low-degree polynomial threshold functions (PTFs) for decision lists, motivated by applications in computational learning theory. As an intermediate step in their PTF constructions, they constructed low- $L_1$ -weight, low-degree polynomials that approximate the OR function on *all* Boolean inputs. Our construction of lower-weight, lower-degree polynomials that approximate the OR function on low Hamming weight inputs is inspired by and builds on Servedio et al.’s construction of approximations that are accurate on all Boolean inputs.

The proof of our lower bound is inspired by recent work that has established new approximate degree lower bounds via the construction of dual solutions to certain linear programs. In particular, Sherstov [30] showed that approximate degree and PTF degree behave roughly multiplicatively under function composition, while Bun and Thaler [5] gave a refinement of Sherstov’s method in order to resolve the approximate degree of the two-level AND-OR tree, and also gave an explicit dual witness for the approximate degree of any symmetric Boolean function. We extend these

lower bounds along two directions: (1) we show degree lower bounds that take into account the  $L_1$ -weight of the coefficient vector of the approximating polynomial, and (2) our lower bounds hold even when we only require the approximation to be accurate on inputs of low Hamming weight, while prior work only considered approximations that are accurate on all Boolean inputs.

Some prior work has studied the degree of polynomials that point-wise approximate *partial* Boolean functions [32, 33]. Here, a function  $f : Y \rightarrow \mathbb{R}$  is said to be partial if its domain  $Y$  is a strict subset of  $\{-1, 1\}^d$ , and a polynomial  $p$  is said to  $\epsilon$ -approximate  $f$  if

1.  $|f(x) - p(x)| \leq \epsilon$  for all  $x \in Y$ , and
2.  $|p(x)| \leq 1 + \epsilon$  for all  $x \in \{-1, 1\}^d \setminus Y$ .

In contrast, our lower bounds apply even in the absence of Condition 2, i.e., when  $p(x)$  is allowed to take arbitrary values on inputs in  $\{-1, 1\}^d \setminus Y$ .

Finally, while our motivation is private data release, our approximation theoretic results are similar in spirit to recent work of Long and Servedio [25], who are motivated by applications in computational learning theory. Long and Servedio consider halfspaces  $h$  defined on inputs of small Hamming weight, and (using different techniques very different from ours) give upper and lower bounds on the weight of these halfspaces when represented as linear threshold functions.

**Organization.** In Section 3, we describe our private online algorithm and show that it yields the claimed accuracy given the existence of sufficiently low-weight polynomials that approximate the  $d$ -variate OR function on inputs of low Hamming weight. The results of this section are a combination of known techniques in differential privacy [17, 19, 28] and learning theory (see e.g., [24]). Readers familiar with these literatures may prefer to skip Section 3 on first reading. In Section 4, we give our polynomial approximations to the OR function, both on low Hamming weight Boolean inputs and on all Boolean inputs. Finally, in Section 5, we state and prove our lower bounds for polynomial approximations to the OR function on restricted inputs.

## 2. PRELIMINARIES

### 2.1 Differentially Private Sanitizers

Let a *database*  $D \in (\{0, 1\}^d)^n$  be a collection of  $n$  rows  $x^{(1)}, \dots, x^{(n)}$  from a *data universe*  $\{0, 1\}^d$ . We say that two databases  $D, D' \in (\{0, 1\}^d)^n$  are *adjacent* if they differ only on a single row, and we denote this by  $D \sim D'$ .

Let  $\mathcal{A} : (\{0, 1\}^d)^n \rightarrow \mathcal{R}$  be an algorithm that takes a database as input and outputs some data structure in  $\mathcal{R}$ . We are interested in algorithms that satisfy *differential privacy*.

**DEFINITION 2.1 (DIFFERENTIAL PRIVACY [10]).** *An algorithm  $\mathcal{A} : (\{0, 1\}^d)^n \rightarrow \mathcal{R}$  is  $(\epsilon, \delta)$ -differentially private if for every two adjacent databases  $D \sim D' \in (\{0, 1\}^d)^n$  and every subset  $S \subseteq \mathcal{R}$ ,*

$$\Pr[\mathcal{A}(D) \in S] \leq e^\epsilon \Pr[\mathcal{A}(D') \in S] + \delta.$$

Since a sanitizer that always outputs  $\perp$  satisfies Definition 2.1, we focus on sanitizers that are accurate. In particular, we are interested in sanitizers that give accurate

answers to *counting queries*. A counting query is defined by a boolean predicate  $q : \{0, 1\}^d \rightarrow \{0, 1\}$ . Abusing notation, we define the evaluation of the query  $q$  on a database  $D \in (\{0, 1\}^d)^n$  to be  $q(D) = \frac{1}{n} \sum_{i=1}^n q(x^{(i)})$ . Note that the value of a counting query is in  $[0, 1]$ . We use  $\mathcal{Q}$  to denote a set of counting queries.

For the purposes of this work, we assume that the range of  $\mathcal{A}$  is simply  $\mathbb{R}^{|\mathcal{Q}|}$ . That is,  $\mathcal{A}$  outputs a list of real numbers representing answers to each of the specified queries.

**DEFINITION 2.2 (ACCURACY).** *The output of  $\mathcal{A}(D)$ ,  $a = (a_q)_{q \in \mathcal{Q}}$ , is  $\alpha$ -accurate for the query set  $\mathcal{Q}$  if*

$$\forall q \in \mathcal{Q}, |a_q - q(D)| \leq \alpha$$

*A sanitizer is  $(\alpha, \beta)$ -accurate for the query set  $\mathcal{Q}$  if for every database  $D$ ,  $\mathcal{A}(D)$  outputs  $a$  such that with probability at least  $1 - \beta$ , the output  $a$  is  $\alpha$ -accurate for  $\mathcal{Q}$ , where the probability is taken over the coins of  $\mathcal{A}$ .*

We remark that the definition of both differential privacy and  $(\alpha, \beta)$ -accuracy extend straightforwardly to the online setting. Here the algorithm receives a sequence of  $\ell$  (possibly adaptively chosen) queries from  $\mathcal{Q}$  and must give an answer to each before seeing the rest of the sequence. Here we require that with probability at least  $1 - \beta$ , every answer output by the algorithm is within  $\pm\alpha$  of the true answer on  $D$ . See e.g., [19] for an elaborate treatment of the online setting.

### 2.2 Query Function Families

Given a set of queries of interest,  $\mathcal{Q}$  (e.g., all marginal queries), we think of the database  $D$  as specifying a function  $f_D$  mapping queries  $q$  to their answers  $q(D)$ . We now describe this transformation more formally:

**DEFINITION 2.3 (Q-FUNCTION FAMILY).** *Let*

$$\mathcal{Q} = \{q_y\}_{y \in Y_{\mathcal{Q}} \subseteq \{-1, 1\}^m}$$

*be a set of counting queries on a data universe  $\{0, 1\}^d$ , where each query is indexed by an  $m$ -bit string. We define the index set of  $\mathcal{Q}$  to be the set  $Y_{\mathcal{Q}} = \{y \in \{-1, 1\}^m \mid q_y \in \mathcal{Q}\}$ .*

*We define the  $\mathcal{Q}$ -function family*

$$\mathcal{F}_{\mathcal{Q}} = \{f_x : \{-1, 1\}^m \rightarrow [0, 1]\}_{x \in \{0, 1\}^d}$$

*as follows: For every possible database row  $x \in \{0, 1\}^d$ , the function  $f_x : \{-1, 1\}^m \rightarrow [0, 1]$  is defined as  $f_x(y) = q_y(x)$ . Given a database  $D \in (\{0, 1\}^d)^n$  we define the function  $f_{\mathcal{Q}, D} : \{-1, 1\}^m \rightarrow [0, 1]$  where  $f_{\mathcal{Q}, D}(y) = \frac{1}{n} \sum_{i=1}^n f_{x^{(i)}}(y)$ . When  $\mathcal{Q}$  is clear from context we will drop the subscript  $\mathcal{Q}$  and simply write  $f_x$ ,  $f_D$ , and  $\mathcal{F}$ .*

When  $\mathcal{Q}$  is the set of all monotone  $k$ -way disjunctions on a database  $D \in (\{0, 1\}^d)^n$ , the queries are defined by sets  $S \subseteq [d]$ ,  $|S| \leq k$ . In this case, we represent each query by the  $d$ -bit  $-1/1$  indicator vector  $y_S$  of the set  $S$ , where  $y_S(i) = -1$  if and only if  $i \in S$ . Thus,  $y_S$  has at most  $k$  entries that are  $-1$ . Hence, we can take  $m = d$  and  $Y_{\mathcal{Q}} = \{y \in \{-1, 1\}^d \mid \sum_{j=1}^d \mathbf{1}_{\{y_j = -1\}} \leq k\}$ .

### 2.3 Low-Weight Polynomial Approximations

Given an  $m$ -variate real polynomial  $p : \{-1, 1\}^m \rightarrow \mathbb{R}$ ,

$$p(y) = \sum_{S \subseteq [m]} c_S \cdot \prod_{i \in S} y_i,$$

we define the degree, weight  $w(\cdot)$  and *non-constant weight*  $w^*(\cdot)$  of the polynomial as follows:

$$\begin{aligned} \deg(p) &:= \max\{|S| : S \subseteq [m], c_S \neq 0\}, \\ w(p) &:= \sum_{S \subseteq [m]} |c_S|, \text{ and} \\ w^*(p) &:= \sum_{S \subseteq [m], S \neq \emptyset} |c_S|. \end{aligned}$$

We use  $\binom{[m]}{\leq t}$  to denote  $\{S \subseteq [m] \mid |S| \leq t\}$  and  $\binom{m}{\leq t} = \left| \binom{[m]}{\leq t} \right| = \sum_{j=0}^t \binom{m}{j}$ .

We will attempt to approximate the functions

$$f_x : \{-1, 1\}^m \rightarrow \{0, 1\}$$

on all the indices in  $Y_{\mathcal{Q}}$  by a family of polynomials with low degree and low weight. Formally and more generally:

**DEFINITION 2.4. [RESTRICTED APPROXIMATION BY POLYNOMIALS]** *Given a function  $f : Y \rightarrow \mathbb{R}$ , where  $Y \subseteq \mathbb{R}^m$ , and a subset  $Y' \subseteq Y$ , we denote the restriction of  $f$  to  $Y'$  by  $f|_{Y'}$ . Given an  $m$ -variate real polynomial  $p$ , we say that  $p$  is a  $\gamma$ -approximation to the restriction  $f|_{Y'}$ , if  $|f(y) - p(y)| \leq \gamma \forall y \in Y'$ . Notice there is no restriction whatsoever placed on  $p(y)$  for  $y \in Y \setminus Y'$ .*

*Given a family of  $m$ -variate functions*

$$\mathcal{F} = \{f_x : Y \rightarrow \mathbb{R}\}_{x \in \{0,1\}^d},$$

where  $Y \subseteq \mathbb{R}^m$ , a set  $Y' \subseteq Y$  we use  $\mathcal{F}|_{Y'} = \{f_x|_{Y'}\}_{x \in \{0,1\}^d}$  to denote the family of restricted functions. Given a family  $\mathcal{P}$  of  $m$ -variate real polynomials, we say that the family  $\mathcal{P}$  is a  $\gamma$ -approximation to the family  $\mathcal{F}|_{Y'}$  if for every  $x \in \{0,1\}^d$ , there exists  $p_x \in \mathcal{P}$  that is a  $\gamma$ -approximation to  $f_x|_{Y'}$ .

Let  $H_{m,k} = \{x \in \{-1, 1\}^m : \sum_{i=1}^m (1 - x_i)/2 \leq k\}$  denote the set of inputs of Hamming weight at most  $k$ . We view the  $d$  variate OR function,  $\text{OR}_d$  as mapping inputs from  $\{-1, 1\}^d$  to  $\{-1, 1\}$ , with the convention that  $-1$  is TRUE and  $1$  is FALSE. Let  $\mathcal{P}_{t,W}(m)$  denote the family of all  $m$ -variate real polynomials of degree  $t$  and weight  $W$ . For the upper bound, we will show that for certain small values of  $t$  and  $W$ , the family  $\mathcal{P}_{t,W}(d)$  is a  $\gamma$ -approximation to the family of all disjunctions restricted to  $H_{d,k}$ .

**FACT 2.5.** *If  $\mathcal{Q}$  is the set of all monotone  $k$ -way disjunctions on a database  $D \in (\{0,1\}^d)^n$ ,  $\mathcal{F}$  is its function family, and  $Y = H_{d,k}$  is its index set, then  $\mathcal{P}_{t,W}(d)$  is a  $\gamma$ -approximation to the restriction  $\mathcal{F}|_Y$  if and only if there is a degree  $t$  polynomial of weight  $O(W)$  that  $\gamma$ -approximates  $\text{OR}_d|_{H_{d,k}}$ .*

The fact follows easily by observing that for any  $x \in \{0,1\}^d$ ,  $y \in \{-1, 1\}^d$ ,

$$f_x(y) = \bigvee_{i \in x} \mathbf{1}_{\{y_i = -1\}} = \frac{1 - \text{OR}_d(y_1^{x_1}, \dots, y_d^{x_d})}{2}.$$

For the lower bound, we will show that any collection of polynomials with small weight that is a  $\gamma$ -approximation to the family of disjunctions restricted to  $H_{m,k}$  should have large degree. We need the following definitions:

**DEFINITION 2.6 (APPROXIMATE DEGREE).** *Given a function  $f : Y \rightarrow \mathbb{R}$ , where  $Y \subseteq \mathbb{R}^m$ , the  $\gamma$ -approximate degree of  $f$  is*

$$\deg_{\gamma}(f) := \min\{d : \exists \text{ real polynomial } p \text{ that is a } \gamma\text{-approximation to } f, \deg(p) = d\}.$$

Analogously, the  $(\gamma, W)$ -approximate degree of  $f$  is

$$\deg_{(\gamma, W)}(f) := \min\{d : \exists \text{ real polynomial } p \text{ that is a } \gamma\text{-approximation to } f, \deg(p) = d, w(p) \leq W\}.$$

It is clear that  $\deg_{\gamma}(f) = \deg_{(\gamma, \infty)}(f)$ .

We let  $w^*(f, t)$  denote the degree- $t$  non-constant margin weight of  $f$ , defined to be:

$$\begin{aligned} w^*(f, t) &:= \min\{w^*(p) : \exists \text{ real polynomial } p \text{ s.t.} \\ &\deg(p) \leq t, f(y)p(y) \geq 1 \forall y \in Y\}. \end{aligned}$$

The above definitions extend naturally to the restricted function  $f|_{Y'}$ .

Our definition of non-constant margin weight is closely related to the well-studied notion of the degree- $t$  polynomial threshold function (PTF) weight of  $f$  (see e.g., [31]), which is defined as  $\min_p w(p)$ , where the minimum is taken over all degree- $t$  polynomials  $p$  with integer coefficients, such that  $f(x) = \text{sign}(p(x))$  for all  $x \in \{-1, 1\}^d$ . Often, when studying PTF weight, the requirement that  $p$  have integer coefficients is used only to ensure that  $p$  has non-trivial margin, i.e. that  $|p(x)| \geq 1$  for all  $x \in \{-1, 1\}^d$ ; this is precisely the requirement captured in our definition of non-constant margin weight. We choose to work with margin weight because it is a cleaner quantity to analyze using linear programming duality; PTF weight can also be studied using LP duality, but the integrality constraints on the coefficients of  $p$  introduces an integrality gap that causes some loss in the analysis (see e.g., Sherstov [31, Theorem 3.4] and Klauck [23, Section 4.3]).

### 3. PRIVATE DATA RELEASE USING LOW-WEIGHT APPROXIMATIONS

In this section we show that low-weight polynomial approximations imply data release algorithms that provide approximate answers even on small databases. The main goal of this section is to prove the following theorem.

**THEOREM 3.1.** *Given  $\alpha, \beta, \varepsilon, \delta > 0$ , and a family of linear queries  $\mathcal{Q}$  with index set  $Y \subseteq \{-1, 1\}^m$ . Suppose for some  $t \leq m$ ,  $W > 0$ , the family of polynomials  $\mathcal{P}_{t,W}(m)$  ( $\alpha/4$ )-approximates the function family  $\mathcal{F}_{\mathcal{Q}}|_Y$ . Then there exists an  $(\varepsilon, \delta)$ -differentially private online algorithm that is  $(4\alpha, \beta)$ -accurate for any sequence of  $\ell$  (possibly adaptively chosen) queries from  $\mathcal{Q}$  on a database  $D \in (\{0, 1\}^d)^n$ , provided*

$$n \geq \frac{128W \log(\ell/\beta) \log(4/\delta)}{\alpha^2 \varepsilon} \sqrt{\log\left(2 \binom{m}{\leq t} + 1\right)}.$$

The private algorithm has running time that is polynomial in  $(n, \binom{m}{\leq t}, \log W, \log(1/\alpha), \log(1/\beta), \log(1/\varepsilon), \log(1/\delta))$ .

We note that the theorem can be assembled from known techniques in the design and analysis of differentially private

algorithms and online learning algorithms. We include the proof of the theorem here for the sake of completeness, as to our knowledge they do not explicitly appear in the privacy literature.

We construct and analyze the algorithm in two steps. First, we use standard arguments to show that the non-private multiplicative weights algorithm can be used to construct a suitable online learning algorithm for  $f_{\mathcal{Q},D}$  whenever  $f_{\mathcal{Q},D}$  can be approximated by a low-weight, low-degree polynomial. Here, a suitable online learning algorithm is one that fits into the IDC framework of Gupta et al. [17]. We then apply the generic conversion from IDCs to differentially private online algorithms [17, 19, 28] to obtain our algorithm.

### 3.1 IDCs

We start by providing the relevant background on the *iterative database construction* framework. An IDC will maintain a sequence of functions  $f_D^{(1)}, f_D^{(2)}, \dots$  that give increasingly good approximations to the  $f_{\mathcal{Q},D}$ . In our case, these functions will be low-degree polynomials. Moreover, the mechanism produces the next approximation in the sequence by considering only one query  $y^{(t)}$  that “distinguishes” the real database in the sense that  $|f^{(t)}(y^{(t)}) - f_D(y^{(t)})|$  is large.

DEFINITION 3.2 (IDC [17, 19, 28]). *Let*

$$\mathcal{Q} = \{q_y\}_{y \in Y_{\mathcal{Q}} \subseteq \{-1,1\}^m}$$

*be a family of counting queries indexed by  $m$ -bit strings. Let  $\mathbf{U}$  be an algorithm mapping a function  $f: Y_{\mathcal{Q}} \rightarrow \mathbb{R}$ , a query  $y \in Y_{\mathcal{Q}}$ , and a real number  $a$  to a new function  $f'$ . Let  $D \in (\{0,1\}^d)^n$  be a database and  $\alpha > 0$  be a parameter. Consider the following game with an adversary. Let  $f_D^{(1)}$  be some function. In each round  $t = 1, 2, \dots$ :*

1. *The adversary chooses a query  $y^{(t)} \in Y_{\mathcal{Q}}$  (possibly depending on  $f_D^{(t)}$ ).*
2. *If  $|f_D^{(t)}(y^{(t)}) - f_D(y^{(t)})| > \alpha$ , then we say that the algorithm has made a mistake.*
3. *If the algorithm made a mistake, then it receives a value  $a^{(t)} \in \mathbb{R}$  such that  $|a^{(t)} - f_D(y^{(t)})| \leq \alpha/2$  and computes a new function  $f_D^{(t+1)} = \mathbf{U}(f_D^{(t)}, y^{(t)}, a^{(t)})$ . Otherwise let  $f_D^{(t+1)} = f_D^{(t)}$ .*

*If the number of rounds  $t$  in which the algorithm makes a mistake is at most  $B$  for every adversary, then  $\mathbf{U}$  is a iterative database construction for  $\mathcal{Q}$  with mistake bound  $B$ .*

THEOREM 3.3 (VARIANT OF [17]). *For any  $\alpha > 0$ , and any family of queries  $\mathcal{Q}$ , if there is an iterative database construction for  $\mathcal{Q}$  with mistake bound  $B$ , then there is an  $(\varepsilon, \delta)$ -differentially private online algorithm that is  $(4\alpha, \beta)$ -accurate for any sequence of  $\ell$  (possibly adaptively chosen) queries from  $\mathcal{Q}$  on a database  $D \in (\{0,1\}^d)^n$ , so long as*

$$n \geq \frac{32\sqrt{B} \log(\ell/\beta) \log(4/\delta)}{\alpha\varepsilon}.$$

*Moreover, if the iterative database construction,  $\mathbf{U}$ , runs in time  $T_{\mathbf{U}}$ , then the private algorithm has running time  $\text{poly}(n, T_{\mathbf{U}}, \log(1/\alpha), \log(1/\beta), \log(1/\varepsilon), \log(1/\delta))$  per query.*

The IDC we will use is specified in Algorithm 1. The IDC will use approximations  $f_D^{(t)}$  in the form of low-degree polynomials of low-weight, and thus we need to specify how to represent such a function. Specifically, we will represent a polynomial  $p$  as a vector  $\bar{p}$  of length  $2\binom{m}{\leq t} + 1$  with only non-negative entries. For each coefficient  $S \in \binom{[m]}{\leq t}$ , the vector will have two components  $\bar{p}_S, \bar{p}_{-S}$ . (Recall that  $\binom{[m]}{\leq t} := \{S \subseteq [m] \mid |S| \leq t\}$ .) Intuitively these two entries represent the positive part and negative part of the coefficient  $c_S$  of  $p$ . There will also be an additional entry  $\bar{p}_0$  that is used to ensure that the  $L_1$ -norm of the vector is exactly 1. Given a polynomial  $p \in \mathcal{P}_{t,W}$  with coefficients  $(c_S)$ , we can construct this vector by setting

$$\bar{p}_S = \frac{\max\{0, c_S\}}{W} \quad \bar{p}_{-S} = \frac{\max\{0, -c_S\}}{W}$$

and choosing  $\bar{p}_0$  so that  $\|\bar{p}\|_1 = 1$ . Observe that  $\bar{p}_0$  can always be set appropriately since the weight of  $p$  is at most  $W$ .

Similarly, we want to associate queries with vectors so that we can replace the evaluation of the polynomial  $p$  on a query  $y$  with the inner product  $W\langle \bar{p}, \bar{y} \rangle$ . We can do so by defining the vector  $\bar{y}$  of length  $2\binom{m}{\leq t} + 1$  such that  $\bar{y}_0 = 0$ ,  $\bar{y}_S = \prod_{i \in S} y_i$  and  $\bar{y}_{-S} = -\prod_{i \in S} y_i$ .

FACT 3.4. *For every  $m$ -variate polynomial  $p$  of degree at most  $t$  and weight at most  $W$ , and every query  $y \in \{-1, 1\}^m$ ,  $W\langle \bar{p}, \bar{y} \rangle = p(y)$ .*

---

**Algorithm 1** The Multiplicative Weights Algorithm for Low-Weight Polynomials.

---

$\mathbf{U}_{\alpha}^{MW}(\bar{p}^{(t)}, y^{(t)}, a^{(t)})$ :

**Let**  $\eta \leftarrow \alpha/4W$ .

**If**  $p^{(t)} = \emptyset$  **then:** output

$$\bar{p}^{(t)} = \frac{1}{2\binom{m}{\leq t} + 1} (1, \dots, 1)$$

(representing the constant 0 polynomial).

**Else if:**  $a^{(t)} < W\langle \bar{p}^{(t)}, \bar{y}^{(t)} \rangle$

**Let**  $\bar{r}^{(t)} = \bar{y}^{(t)}$

**Else:**

**Let**  $\bar{r}^{(t)} = -\bar{y}^{(t)}$

**Update:** For all  $I \in \left\{0, \binom{[m]}{\leq t}, -\binom{[m]}{\leq t}\right\}$  let

$$\bar{p}_I^{(t+1)} \leftarrow \exp(-\eta \bar{r}_I^{(t)}) \cdot \bar{p}_I^{(t)}$$

$$\bar{p}^{(t+1)} \leftarrow \frac{\bar{p}^{(t+1)}}{\|\bar{p}^{(t+1)}\|_1}$$

**Output**  $\bar{p}^{(t+1)}$ .

---

We summarize the properties of the multiplicative weights algorithm in the following theorem:

THEOREM 3.5. *For any  $\alpha > 0$ , and any family of linear queries  $\mathcal{Q}$  if  $\mathcal{P}_{t,W}$   $(\alpha/4)$ -approximates the restriction  $\mathcal{F}|_Y$  then Algorithm 1 is an iterative database construction for  $\mathcal{Q}$  with mistake bound  $B$  for*

$$B = B(W, m, t, \alpha) = \frac{16W^2 \log\left(2\binom{m}{\leq t} + 1\right)}{\alpha^2}.$$

Moreover,  $\mathbf{U}$  runs in time  $\text{poly}\left(\binom{m}{\leq t}, \log W, \log(1/\alpha)\right)$ .

PROOF. Let  $D \in (\{0, 1\}^d)^n$  be any database. For every round  $t$  in which  $\mathbf{U}$  makes a mistake, we consider the tuple  $(\bar{p}^{(t)}, y^{(t)}, \hat{a}^{(t)})$  representing the information used to update the approximation in round  $t$ . In order to bound the number of mistakes, it will be sufficient to show that after  $B \leq 16W^2 \log(2\binom{m}{\leq t} + 1)/\alpha^2$ , the vector  $\bar{p}^{(t)}$  is such that

$$\forall y \in Y_{\mathcal{Q}}, |W\langle \bar{p}^{(t)}, \bar{y} \rangle - f_D(y)| \leq \alpha.$$

That is, after making  $B$  mistakes  $\bar{p}^{(t)}$  represents a polynomial that approximates  $f_D$  on every query, and thus there can be no more than  $B$  makes.

First, we note that there always exists a polynomial  $p_D \in \mathcal{P}_{t,W}$  such that

$$\forall y \in Y_{\mathcal{Q}}, |p_D(y) - f_D(y)| \leq \frac{\alpha}{4}. \quad (1)$$

The assumption of our theorem is that for every  $x^{(i)} \in D$ , there exists  $p_{x^{(i)}} \in \mathcal{P}_{t,W}$  such that

$$\forall y \in Y_{\mathcal{Q}}, |p_{x^{(i)}}(y) - f_{x^{(i)}}(y)| \leq \frac{\alpha}{4}.$$

Thus, since  $f_D = \frac{1}{n} \sum_{i=1}^n f_{x^{(i)}}$ , the polynomial

$$p_D = \frac{1}{n} \sum_{i=1}^n p_{x^{(i)}}$$

will satisfy (1). Note that  $p_D \in \mathcal{P}_{t,W}$ , thus if we represent  $p_D$  as a vector,

$$\forall y \in Y_{\mathcal{Q}}, |W\langle \bar{p}_D, \bar{y} \rangle - f_D(y)| \leq \frac{\alpha}{4}.$$

Given the existence of  $\bar{p}_D$ , we will define a potential function capturing how far  $\bar{p}^{(t)}$  is from  $\bar{p}_D$ . Specifically, we define

$$\Psi_t := KL(\bar{p}_D || \bar{p}^{(t)}) = \sum_{I \in \{0, \binom{[m]}{\leq t}, -\binom{[m]}{\leq t}\}} \bar{p}_{D,I} \log \left( \frac{\bar{p}_{D,I}}{\bar{p}_I^{(t)}} \right)$$

to be the KL divergence between  $\bar{p}_D$  and the current approximation  $\bar{p}^{(t)}$ . Note that the sum iterates over all  $2\binom{m}{\leq t} + 1$  indices in  $\bar{p}$ . We have the following fact about KL divergence.

FACT 3.6. For all  $t$ :  $\Psi_t \geq 0$ , and  $\Psi_0 \leq \log(2\binom{m}{\leq t} + 1)$ .

We will argue that after each mistake the potential drops by at least  $\alpha^2/16W^2$ . Note that the potential only changes in rounds where a mistake was made. Because the potential begins at  $\log(2\binom{m}{\leq t} + 1)$ , and must always be non-negative, we know that there can be at most  $B(\alpha) \leq 16W^2 \log(2\binom{m}{\leq t} + 1)/\alpha^2$  mistakes before the algorithm outputs a (vector representation of) a polynomial that approximates  $f_D$  on  $Y_{\mathcal{Q}}$ .

The following lemma is standard in the analysis of algorithms based on multiplicative-weights.

LEMMA 3.7.

$$\Psi_t - \Psi_{t+1} \geq \eta \left( \langle \bar{p}^{(t)}, r^{(t)} \rangle - \langle \bar{p}_D, r^{(t)} \rangle \right) - \eta^2$$

PROOF.

$$\begin{aligned} \Psi_t - \Psi_{t+1} &= \sum_{I \in \{0, \binom{[m]}{\leq t}, -\binom{[m]}{\leq t}\}} \bar{p}_{D,I} \log \left( \frac{\bar{p}_I^{(t+1)}}{\bar{p}_I^{(t)}} \right) \\ &= -\eta \langle \bar{p}_D, r^{(t)} \rangle \\ &\quad - \log \left( \sum_{I \in \{0, \binom{[m]}{\leq t}, -\binom{[m]}{\leq t}\}} \exp(-\eta r_I^{(t)}) \bar{p}_I^{(t)} \right) \\ &\geq -\eta \langle \bar{p}_D, r^{(t)} \rangle \\ &\quad - \log \left( \sum_{I \in \{0, \binom{[m]}{\leq t}, -\binom{[m]}{\leq t}\}} \bar{p}_I^{(t)} (1 + \eta^2 - \eta r_I^{(t)}) \right) \\ &\geq \eta \left( \langle \bar{p}^{(t)}, r^{(t)} \rangle - \langle \bar{p}_D, r^{(t)} \rangle \right) - \eta^2 \end{aligned}$$

□

The rest of the proof now follows easily. By the conditions of an iterative database construction algorithm,  $|\hat{a}^{(t)} - f_D(y^{(t)})| \leq \alpha/2$ . Hence, for each  $t$  such that  $|W\langle \bar{p}^{(t)}, y^{(t)} \rangle - f_D(y^{(t)})| \geq \alpha$ , we also have that  $W\langle \bar{p}^{(t)}, y^{(t)} \rangle > f_D(y^{(t)})$  if and only if  $W\langle \bar{p}^{(t)}, y^{(t)} \rangle > \hat{a}^{(t)}$ .

In particular, if  $r^{(t)} = y^{(t)}$ , then

$$W\langle \bar{p}^{(t)}, y^{(t)} \rangle - W\langle \bar{p}_D, y^{(t)} \rangle \geq \alpha/2.$$

Similarly, if  $r^{(t)} = -y^{(t)}$ , then

$$W\langle \bar{p}_D, y^{(t)} \rangle - W\langle \bar{p}^{(t)}, y^{(t)} \rangle \geq \alpha.$$

Here we have utilized the fact that  $|p_D(y) - f_D(y)| \leq \alpha/4$ . Therefore, by Lemma 3.7 and the fact that  $\eta = \alpha/4W$ :

$$\begin{aligned} \Psi_t - \Psi_{t+1} &\geq \frac{\alpha}{4W} \left( \langle \bar{p}^{(t)}, r^{(t)} \rangle - \langle \bar{p}_D, r^{(t)} \rangle \right) - \frac{\alpha^2}{16W^2} \\ &\geq \frac{\alpha}{4W} \left( \frac{\alpha}{2W} \right) - \frac{\alpha^2}{16W^2} = \frac{\alpha^2}{16W^2} \end{aligned}$$

Theorem 3.1 follows immediately from Theorems 3.3 and 3.5.

## 4. UPPER BOUNDS

Fact 2.5 and Theorem 3.1 show that in order to develop a differentially private mechanism that can release all  $k$ -way marginals of a database, it is sufficient to construct a low-weight polynomial that approximates the  $\text{OR}_d$ , on all Boolean inputs of Hamming weight at most  $k$ . This is the purpose to which we now turn.

The  $\text{OR}_d$  function is easily seen to have an exact polynomial representation of constant weight and degree  $d$  (see Fact 4.3 below); however, an approximation with smaller degree may be achieved at the expense of larger weight. The best known weight-degree tradeoff, implicit in the work of Servedio et al. [29], can be stated as follows: there exists a polynomial  $p$  of degree  $t$  and weight  $(d \log(1/\gamma)/t)^{(d(\log 1/\gamma)^2/t)}$  that  $\gamma$ -approximates the function  $\text{OR}_d$  on all Boolean inputs, for every  $t$  larger than  $\sqrt{d} \log(1/\gamma)$ . Setting the degree  $t$  to be  $O(d/\log^{0.99} d)$  yields a polynomial of weight at most  $d^{0.01}$  that approximates the  $\text{OR}_d$  function on all Boolean inputs to any desired constant accuracy. On the other hand, Lemma 8 of [29] can be shown to imply that any polynomial of weight

$W$  that  $1/3$ -approximates the  $\text{OR}_d$  function requires degree  $\Omega(d/\log W)$ , essentially matching the  $O(d/\log^{.99} d)$  upper bound of Servedio et al. when  $W = d^{\Omega(1)}$ .

However, in order to privately release  $k$ -way marginals, we have shown that it suffices to construct polynomials that are accurate *only* on inputs of low Hamming weight. In this section, we give a construction that achieves significantly improved weight degree trade-offs in this setting. In the next section, we demonstrate the tightness of our construction by proving matching lower bounds.

We construct our approximations by decomposing the  $d$ -variate OR function into an OR of OR's, which is the same approach taken by Servedio et al. [29]. Here, the outer OR has fan-in  $m$  and the inner OR has fan-in  $d/m$ , where the subsequent analysis will determine the appropriate choice of  $m$ . In order to obtain an approximation that is accurate on all Boolean inputs, Servedio et al. approximate the outer OR using a transformation of the Chebyshev polynomials of degree  $\sqrt{m}$ , and compute each of the inner OR's exactly.

For  $k \ll \log^2 d$ , we are able to substantially reduce the degree of the approximating polynomial, relative to the construction of Servedio et al., by leveraging the fact that we are interested in approximations that are accurate only on inputs of Hamming weight at most  $k$ . Specifically, we are able to approximate the outer OR function using a polynomial of degree only  $\sqrt{k}$  rather than  $\sqrt{m}$ , and argue that the weight of the resulting polynomial is still bounded by a polynomial in  $d$ .

We now proceed to prove the main lemmas. For the sake of intuition, we begin with weight-degree tradeoffs in the simpler setting in which we are concerned with approximating the  $\text{OR}_d$  function over all Boolean inputs. The following lemma, proved below for completeness, is implicit in the work of [29].

**LEMMA 4.1.** *For every  $\gamma > 0$  and  $m \in [d]$ , there is a polynomial of degree  $t = O((d/\sqrt{m}) \log(1/\gamma))$  and weight  $W = m^{O(\sqrt{m} \log(1/\gamma))}$  that  $\gamma$ -approximates the  $\text{OR}_d$  function.*

Our main contribution in this section is the following lemma that gives an improved polynomial approximation to the  $\text{OR}_d$  function restricted to inputs of low Hamming weight.

**LEMMA 4.2.** *For every  $\gamma > 0$ ,  $k < d$  and  $m \in [d] \setminus [k]$ , there is a polynomial of degree  $t = O(d\sqrt{k} \log(1/\gamma)/m)$  and weight  $W = m^{O(\sqrt{k} \log(1/\gamma))}$  that  $\gamma$ -approximates the  $\text{OR}_d$  function restricted to inputs of Hamming weight at most  $k$ .*

For any constant  $\gamma$ , one may take  $m = d^{O(1/\sqrt{k})}$  in the lemma (here the choice of constant depends on the constants in Fact 4.4) and obtain a polynomial of degree  $d^{1-\Omega(1/\sqrt{k})}$  and weight  $d^{0.01}$ .

Our constructions use the following basic facts.

**FACT 4.3.** *The real polynomial  $p_d : \{-1, 1\}^d \rightarrow \mathbb{R}$*

$$p_d(x) = 2 \left( \sum_{S \subseteq [d]} 2^{-d} \prod_{i \in S} x_i \right) - 1 = 2 \prod_{i=1}^d \left( \frac{1+x_i}{2} \right) - 1$$

*computes  $\text{OR}_d(x)$  and has weight  $w(p_d) \leq 3$ .*

**FACT 4.4.** *[see e.g., [34]] For every  $k \in \mathbb{N}$  and  $\gamma > 0$ , there exists a univariate real polynomial  $p = \sum_{i=0}^{t_k} c_i x^i$  of degree  $t_k$  such that*

1.  $t_k = O(\sqrt{k} \log(1/\gamma))$ ,
2. for every  $i \in [t_k]$ ,  $|c_i| \leq 2^{O(\sqrt{k} \log(1/\gamma))}$ ,
3.  $p(0) = 0$ , and
4. for every  $x \in [2k]$ ,  $|p(x) - 1| \leq \gamma/2$ .

**PROOF OF LEMMA 4.1.** We can compute  $\text{OR}_d(y)$  as a disjunction of disjunctions by partitioning the inputs  $y_1, \dots, y_d$  into blocks of size  $d/m$  and computing:

$$\text{OR}_m(\text{OR}_{d/m}(y_1, \dots, y_{d/m}), \dots, \text{OR}_{d/m}(y_{d-d/m+1}, \dots, y_d)).$$

In order to approximately compute  $\text{OR}_d(y)$ , we compute the inner disjunctions exactly using the polynomial  $p_{d/m}$  given in Fact 4.3 and approximate the outer disjunction using the polynomial from Fact 4.4. Let

$$Z(y) = p_{d/m}(y_1, \dots, y_{d/m}) + \dots + p_{d/m}(y_{d-d/m+1}, \dots, y_d).$$

Setting  $k = m$  in Fact 4.4, let  $q_m$  be the resulting polynomial of degree  $O(\sqrt{m} \log(1/\gamma))$  and weight  $O(m^{\sqrt{m} \log(1/\gamma)})$ . Our final polynomial is

$$1 - 2q_m(m - Z(y)).$$

Note that  $m - Z(y)$  takes values in  $\{0, \dots, m\}$  and is 0 exactly when all inputs  $y_1, \dots, y_d$  are FALSE. It follows that our final polynomial indeed approximates  $\text{OR}_d$  to additive error  $\gamma$  on all Boolean inputs.

We bound the degree and weight of this polynomial in  $y$ . By Fact 4.3, the inner disjunctions are computed exactly using degree  $d/m$  and weight at most 3. Hence, the total degree is  $O(\sqrt{m} \log(1/\gamma) \cdot d/m)$ . To bound the weight, we observe that the outer polynomial  $q_m(\cdot)$  has at most  $T = m^{O(\sqrt{m} \log(1/\gamma))}$  terms where each one has degree at most  $D_{\text{outer}} = O(\sqrt{m} \log(1/\gamma))$  and coefficients of absolute value at most  $c_{\text{outer}} = 2^{O(\sqrt{m} \log(1/\gamma))}$ . Expanding the polynomials for  $Z(y)$ , the weight of each term incurs a multiplicative factor of  $c_{\text{inner}} \leq 3^{D_{\text{outer}}} = 3^{O(\sqrt{m} \log(1/\gamma))}$  so the total weight is at most  $c_{\text{inner}} \cdot c_{\text{outer}} \cdot T = m^{O(\sqrt{m} \log(1/\gamma))}$ .

□

**PROOF OF LEMMA 4.2.** Again we partition the inputs  $y_1, \dots, y_d$  into blocks of size  $d/m$  and view the disjunction as:

$$\text{OR}_m(\text{OR}_{d/m}(y_1, \dots, y_{d/m}), \dots, \text{OR}_{d/m}(y_{d-d/m+1}, \dots, y_d)).$$

Once again, we compute the inner disjunctions exactly using the polynomial from Fact 4.3. Let

$$Z(y) = p_{d/m}(y_1, \dots, y_{d/m}) + \dots + p_{d/m}(y_{d-d/m+1}, \dots, y_d).$$

If the input  $y$  has Hamming weight at most  $k$ , then  $Z(y)$  also takes values in  $\{m, \dots, m - 2k\}$ . Thus, we may approximate the outer disjunction using a polynomial of degree  $O(\sqrt{k} \log(1/\gamma))$  from Fact 4.4. Our final polynomial is:

$$1 - 2q_k(m - Z(y)).$$

The bound on degree and weight may be obtained as in the previous lemma. □

## 4.1 Proof of upper bound theorems

We first present the proof of Theorem 1.3.

PROOF OF THEOREM 1.3. Taking

$$m = O((\log d / \log \log d)^2)$$

in Lemma 4.1 and  $m = d^{O(1/\sqrt{k})}$  in Lemma 4.2, it follows that for some constant  $C > 0$ , the  $d$ -variate disjunction restricted to  $H_{d,k}$  is  $(1/400)$ -approximated by a  $d$ -variate real polynomial of degree  $t$  and weight  $W$  where

$$t = \min \left\{ d^{1 - \frac{1}{C\sqrt{k}}}, \frac{d}{\log^{0.995} d} \right\} \text{ and } W = d^{0.01}.$$

□

PROOF OF THEOREM 1.1. By Theorem 1.3, we have a polynomial  $p$  that  $(1/400)$ -approximates the function  $OR_d|_{H_{d,k}}$ . Moreover,  $p$  has weight  $W < d^{0.01}$  and degree

$$t \leq \min \left\{ d^{1 - \frac{1}{C\sqrt{k}}}, \frac{d}{\log^{0.995} d} \right\}.$$

Thus, by Fact 2.5, we have a family of polynomials  $\mathcal{P}_{t,W}(d)$  that  $(1/400)$ -approximates the function family  $\mathcal{F}_{\mathcal{Q}}|_{H_{d,k}}$ . We have the ingredients needed to apply Theorem 3.1. Taking  $\alpha = 1/100$ ,  $\beta = 1/100$  gives the conclusion. □

Remark 1 in the Introduction follows from using a slightly different choice of  $m$  in Lemma 4.1, namely

$$m = O(\log^2 d / \log^3 \log d).$$

To obtain the summary of the database promised in Remark 2, we request an answer to each of the  $k$ -way marginal queries  $B(1/400)$  times. Doing so, will ensure that we obtain a maximal database update sequence, and it was argued in Section 2.1 that the polynomial resulting from any maximal database update sequence accurately answers every  $k$ -way marginal query. Finally, we obtain a compact summary by randomly choosing  $\tilde{O}(kd^{0.01})$  samples from the normalized coefficient vector of this polynomial to obtain a new sparse polynomial that accurately answers every  $k$ -way marginal query (see e.g. [4]). Our compact summary is this final sparse polynomial.

## 5. LOWER BOUNDS

In this section, we address the general problem of approximating a block-composed function  $G = F(\dots, f(\cdot), \dots)$ , where  $F: \{-1, 1\}^k \rightarrow \{-1, 1\}$ ,  $f: Y \rightarrow \{-1, 1\}$ ,  $Y \subseteq \mathbb{R}^{d/k}$  over inputs restricted to a set  $\mathcal{Y} \subseteq Y^k$  using low-weight polynomials. We give a lower bound on the minimum degree of such polynomials. In our main application,  $G$  will be  $OR_d$ , and  $\mathcal{Y}$  will be the set of all  $d$ -dimensional Boolean vectors of Hamming weight at most  $k$ .

Our proof technique is inspired by the composition theorem lower bounds of [30, Theorem 3.1], where it is shown that the  $\gamma$ -approximate degree of the composed function  $G$  is at least the product of the  $\gamma$ -approximate degree of the outer function and the PTF degree of the inner function. Our main contribution is a generalization of such a composition theorem along two directions: (1) we show degree lower bounds that take into account the L1-norm of the coefficient vector of the approximating polynomial, and (2) our lower bounds hold even when we require the approximation to be accurate only on inputs of low Hamming weight, while prior work only considered approximations that are accurate on all Boolean inputs.

Our main theorem is stated below. In parsing the statement of the theorem, it may be helpful to think of  $G = OR_d$ ,  $\mathcal{Y} = H_{d,k}$ , the set of all  $d$ -dimensional Boolean vectors of Hamming weight at most  $k$ ,  $f = OR_{d/k}$ ,  $F = OR_k$ ,  $Y = \{-1, 1\}^{d/k}$ , and  $H = H_{d/k,1}$ . This will be the setting of interest in our main application of the theorem.

**THEOREM 5.1.** *Let  $Y \subseteq \mathbb{R}^{d/k}$  be a finite set and  $\gamma > 0$ . Given  $f: Y \rightarrow \{-1, 1\}$  and  $F: \{-1, 1\}^k \rightarrow \{-1, 1\}$  such that  $\deg_{2,\gamma}(F) = D$ , let  $G: Y^k \rightarrow \{-1, 1\}$  denote the composed function defined by  $G(Y_1, \dots, Y_k) = F(f(Y_1), \dots, f(Y_k))$ . Let  $\mathcal{Y} \subseteq Y^k$ . Suppose there exists  $H \subseteq Y$  such that for every  $(Y_1, \dots, Y_k) \in Y^k \setminus \mathcal{Y}$  there exists  $i \in [k]$  such that  $Y_i \in Y \setminus H$ . Then, for every  $t \in \mathbb{Z}_+$ ,*

$$\deg_{(\gamma,W)}(G|_{\mathcal{Y}}) \geq \frac{1}{2}tD \text{ for every } W \leq \gamma^{2-k}w^*(f|_H, t)^{\frac{D}{2}}.$$

We will derive the following corollary from Theorem 5.1. Theorem 1.4 follows immediately from Corollary 5.2 by considering any  $k = o(\log d)$ .

**COROLLARY 5.2.** *Let  $k \in [d]$ . Then, there exists a universal constant  $C > 0$  such that*

$$\deg_{(1/6,W)}(OR_d|_{H_{d,k}}) = \Omega \left( \frac{d}{\sqrt{k}} \cdot \frac{W^{-\frac{1}{C\sqrt{k}}}}{2^{\sqrt{k}/C}} \right)$$

for every

$$W \leq \frac{1}{6 \cdot 2^k} \left( \frac{d}{k} \right)^{C\sqrt{k}}.$$

**Intuition underlying our proof technique.** Recall that our upper bound in Section 4 worked as follows. We viewed  $OR_d$  as an “OR of ORs”, and we approximated the outer OR with a polynomial  $p$  of degree  $\deg_{\text{outer}}$  chosen to be as small as possible, and composed  $p$  with a low-weight but high-degree polynomial computing each inner OR. We needed to make sure the weight  $W_{\text{inner}}$  of the inner polynomials was very low, because the composition step potentially blows the weight up to roughly  $W_{\text{inner}}^{\deg_{\text{outer}}}$ . As a result, the inner polynomials had to have very high degree, to keep their weight low.

Intuitively, we construct a dual solution to a certain linear program that captures the intuition that any low-weight, low-degree polynomial approximation to  $OR_d$  must look something like our primal solution, composing a low-degree approximation to an “outer” OR with low-weight approximations to inner ORs. Moreover, our dual solution formalizes the intuition that the composition step must result in a massive blowup in weight, from  $W_{\text{inner}}$  to roughly  $W_{\text{inner}}^{\deg_{\text{outer}}}$ .

In more detail, our dual construction works by writing  $OR_d$  as an OR of ORs, where the outer OR is over  $k$  variables, and each inner ORs is over  $d/k$  variables. We obtain our dual solution by carefully combining a dual witness  $\Gamma$  to the high approximate degree of the outer OR, with a dual witness  $\psi$  to the fact that any low-degree polynomial with margin at least 1 for each inner OR, must have “large” weight, even if the polynomial must satisfy the margin constraint only on inputs of Hamming weight 0 or 1. This latter condition, that  $\psi$  must witness high non-constant margin weight even if restricted to inputs of Hamming weight 0 or 1, is essential to ensuring that our combined dual witness does not place any “mass” on irrelevant inputs, i.e. those of Hamming weight larger than  $k$ .

## 5.1 Duality Theorems

In the rest of the section, we let  $\chi_S(x) = \prod_{i \in S} x_i$  for any given set  $S \subseteq [d]$ . The question of existence of a weight  $W$  polynomial with small degree that  $\gamma$ -approximates a given function can be expressed as a feasibility problem for a linear program. Now, in order to show the non-existence of such a polynomial, it is sufficient to show infeasibility of the linear program. By duality, this is equivalent to demonstrating the existence of a solution to the corresponding dual program. We begin by summarizing the duality theorems that will be useful in exhibiting this witness.

**THEOREM 5.3** (( $\gamma, W$ )-APPROXIMATE DEGREE). *Fix  $\gamma \geq 0$  and let  $f : Y \rightarrow \{-1, 1\}$  be given for some finite set  $Y \subset \mathbb{R}^d$ . Then,  $\deg_{(\gamma, W)}(f) \geq t + 1$  if and only if there exists a function  $\Psi : Y \rightarrow \mathbb{R}$  such that*

1.  $\sum_{y \in Y} |\Psi(y)| = 1$ ,
2.  $\sum_{y \in Y} \Psi(y)f(y) - W \cdot |\sum_{y \in Y} \Psi(y)\chi_S(y)| > \gamma$  for every  $S \subseteq [d], |S| \leq t$ .

**PROOF.** By definition,  $\deg_{(\gamma, W)}(f) \leq t$  if and only if  $\exists (\lambda_S)_{S \subseteq [d], |S| \leq t}$ :

$$\sum_{S \subseteq [d], |S| \leq t} |\lambda_S| \leq W, \text{ and}$$

$$\left| f(y) - \sum_{S \subseteq [d], |S| \leq t} \lambda_S \chi_S(y) \right| \leq \gamma \quad \forall y \in Y.$$

By Farkas' lemma,  $\deg_{(\gamma, W)}(f) \leq t$  if and only if  $\nexists \Psi : Y \rightarrow \mathbb{R}$  such that

$$\frac{1}{W} \sum_{y \in Y} (f(y)\Psi(y) - \gamma|\Psi(y)|) > \left| \sum_{y \in Y} \chi_S(y)\Psi(y) \right|$$

$\forall S \subseteq [d], |S| \leq t$ .  $\square$

The dual witness that we construct to prove Theorem 5.1 is obtained by combining a dual witness for the large non-constant margin weight of the inner function with a dual witness for the large approximate degree for the outer function. The duality conditions for these are given below. The proof of the duality condition for the case of  $\gamma$ -approximate degree is well-known, so we omit the proof for brevity.

**THEOREM 5.4** ( $\gamma$ -APPROXIMATE DEGREE). [5, 31, 37] *Fix  $\gamma \geq 0$  and let  $f : Y \rightarrow \{-1, 1\}$  be given, where  $Y \subset \mathbb{R}^d$  is a finite set. Then,  $\deg_\gamma(f) \geq t + 1$  if and only if there exists a function  $\Gamma : Y \rightarrow \mathbb{R}$  such that*

1.  $\sum_{y \in Y} |\Gamma(y)| = 1$ ,
2.  $\sum_{y \in Y} \Gamma(y)p(y) = 0$  for every polynomial  $p$  of degree at most  $t$ , and
3.  $\sum_{y \in Y} \Gamma(y)f(y) > \gamma$ .

**THEOREM 5.5** (NON-CONSTANT MARGIN WEIGHT). *Let  $Y \subset \mathbb{R}^d$  be a finite set, let  $f : Y \rightarrow \{1, -1\}$  be a given function and  $w > 0$ . The non-constant margin weight  $w^*(f, t) \geq w$  if and only if there exists a distribution  $\mu : Y \rightarrow [0, 1]$  such that*

1.  $\sum_{y \in Y} \mu(y)f(y) = 0$

2.  $\left| \sum_{y \in Y} \mu(y)f(y)\chi_S(y) \right| \leq \frac{1}{w}$  for every  $S \subseteq [d], |S| \leq t$ .

**PROOF.** Let  $\mathcal{S} = \{S \subseteq [d] : |S| \leq t\}$ ,  $\bar{\mathcal{S}} = \mathcal{S} \setminus \{\emptyset\}$ . By definition,  $w^*(f, t)$  is expressed by the following linear program:

$$\min \sum_{S \in \bar{\mathcal{S}}} |\lambda_S|$$

$$f(y) \sum_{S \in \mathcal{S}} \lambda_S \chi_S(y) \geq 1 \quad \forall y \in Y.$$

The above linear program can be restated as follows:

$$\min \sum_{S \in \bar{\mathcal{S}}} \alpha_S$$

$$\alpha_S + \lambda_S \geq 0 \quad \forall S \in \bar{\mathcal{S}},$$

$$\alpha_S - \lambda_S \geq 0 \quad \forall S \in \bar{\mathcal{S}},$$

$$f(y) \sum_{S \in \mathcal{S}} \lambda_S \chi_S(y) \geq 1 \quad \forall y \in Y, \text{ and}$$

$$\alpha_S \geq 0 \quad \forall S \in \bar{\mathcal{S}}.$$

The dual program is expressed below:

$$\max \sum_y \mu(y)$$

$$u_1(S) + u_2(S) \leq 1 \quad \forall S \in \bar{\mathcal{S}},$$

$$\sum_{y \in Y} \mu(y)f(y)\chi_S(y) + u_1(S) - u_2(S) = 0 \quad \forall S \in \bar{\mathcal{S}},$$

$$\sum_{y \in Y} \mu(y)f(y) = 0,$$

$$\mu(y) \geq 0 \quad \forall y \in Y, \quad u_1(S), u_2(S) \geq 0 \quad \forall S \in \bar{\mathcal{S}}.$$

By standard manipulations, the above dual program is equivalent to

$$\max \sum_y \mu(y)$$

$$\left| \sum_{y \in Y} \mu(y)\chi_S(y)f(y) \right| \leq 1 \quad \forall S \in \bar{\mathcal{S}}$$

$$\sum_{y \in Y} \mu(y)f(y) = 0,$$

$$\mu(y) \geq 0 \quad \forall y \in Y$$

Finally, given a distribution  $\mu'$  satisfying the hypothesis of the theorem, one can obtain a dual solution  $\mu$  to show that  $w^*(f, t) \geq w$  by taking

$$w^{-1} = \max_{S \in \bar{\mathcal{S}}} \left| \sum_{y \in Y} \mu'(y)\chi_S(y)f(y) \right|$$

and setting  $\mu(y) = w\mu'(y) \quad \forall y \in Y$ . In the other direction, if  $w^*(f, t) \geq w$ , then we have a dual solution  $\mu$  satisfying the above dual program such that  $\sum_{y \in Y} \mu(y) = w^*(f, t)$ . By setting  $\mu'(y) = \mu(y)/w^*(f, t) \quad \forall y \in Y$ , we obtain the desired distribution.  $\square$

## 5.2 Proof of Theorem 5.1

Our approach to exhibiting a dual witness as per Theorem 5.3 is to build a dual witness by appropriately combining the dual witnesses for the ‘‘hardness’’ of the inner and outer

functions. Our method of combining the dual witnesses is inspired by the technique of [30, Theorem 3.7].

PROOF OF THEOREM 5.1. Let  $w = w^*(f|_H, t)$ . We will exhibit a dual witness function  $\Psi : \mathcal{Y} \rightarrow \mathbb{R}$  corresponding to Theorem 5.3 for the specified choice of degree and weight. For  $y \in Y^k$ , let  $Y_i = (y_{(i-1)(d/k)+1}, \dots, y_{id/k})$ . By Theorem 5.5, we know that there exists a distribution  $\mu : H \rightarrow \mathbb{R}$  such that

$$\sum_{y \in H} \mu(y) f(y) = 0, \quad (2)$$

$$\left| \sum_{y \in H} \mu(y) f(y) \chi_S(y) \right| \leq \frac{1}{w} \forall S \subseteq \left[ \frac{d}{k} \right], |S| \leq t \quad (3)$$

We set  $\mu(y) = 0$  for  $y \in Y \setminus H$ .

Since  $\deg_{2\gamma}(F) = D$ , by Theorem 5.4, we know that there exists a function  $\Gamma : \{-1, 1\}^k \rightarrow \mathbb{R}$  such that

$$\sum_{x \in \{-1, 1\}^k} |\Gamma(x)| = 1, \quad (4)$$

$$\sum_{x \in \{-1, 1\}^k} \Gamma(x) p(x) = 0$$

for every polynomial  $p$  of degree at most  $D$ , and

$$\sum_{x \in \{-1, 1\}^k} \Gamma(x) F(x) > 2\gamma. \quad (6)$$

Consider the function  $\Psi : Y^k \rightarrow \mathbb{R}$  defined as  $\Psi(y) = 2^k \Gamma(f(Y_1), \dots, f(Y_k)) \prod_{i=1}^k \mu(Y_i)$ . By the hypothesis of the theorem, we know that if  $(Y_1, \dots, Y_k) \in Y^k \setminus \mathcal{Y}$ , then there exists  $i \in [k]$  such that  $Y_i \in Y \setminus H$  and hence  $\mu(Y_i) = 0$  and therefore  $\Psi(Y_1, \dots, Y_k) = 0$ .

1.

$$\begin{aligned} \sum_{y \in \mathcal{Y}} |\Psi(y)| &= \sum_{y \in \mathcal{Y}} 2^k |\Gamma(f(Y_1), \dots, f(Y_k))| \prod_{i=1}^k \mu(Y_i) \\ &= 2^k \mathbb{E}_{y \sim \Phi} (|\Gamma(f(Y_1), \dots, f(Y_k))|) \end{aligned}$$

where  $y \sim \Phi$  denotes  $y$  chosen from the product distribution  $\Phi : Y^k \rightarrow [0, 1]$  defined by  $\Phi(y) = \prod_{i \in [k]} \mu(Y_i)$ . Since  $\sum_{y \in Y} \mu(y) f(y) = 0$ , it follows that if  $Y_i$  is chosen with probability  $\mu(Y_i)$ , then  $f(Y_i)$  is uniformly distributed in  $\{-1, 1\}$ . Consequently,

$$\sum_{y \in \mathcal{Y}} |\Psi(y)| = 2^k \mathbb{E}_{z \sim_U \{-1, 1\}^k} (|\Gamma(z_1, \dots, z_k)|) = 1.$$

The last equality is by using (4).

2. By the same reasoning as above, it follows from (6) that

$$\sum_{y \in \mathcal{Y}} \Psi(y) G(y) = \sum_{z \in \{-1, 1\}^k} \Gamma(z) F(z) > 2\gamma.$$

3. Fix a subset  $S \subseteq [d]$  of size at most  $tD/2$ . Let  $S_i = S \cap \{(i-1)(d/k) + 1, \dots, id/k\}$  for each  $i \in [k]$ . Consequently,  $\chi_S(y) = \prod_{i=1}^k \chi_{S_i}(Y_i)$ .

Now using the Fourier coefficients  $\hat{\Gamma}(T)$  of the function  $\Gamma$ , we can express

$$\Gamma(z_1, \dots, z_k) = \sum_{T \subseteq [k]} \hat{\Gamma}(T) \prod_{i \in T} z_i = \sum_{\substack{T \subseteq [k], \\ |T| \geq D}} \hat{\Gamma}(T) \prod_{i \in T} z_i$$

since  $\hat{\Gamma}(T) = 0$  if  $|T| < D$  by (5). Hence,

$$\Psi(y) = 2^k \sum_{\substack{T \subseteq [k], \\ |T| \geq D}} \hat{\Gamma}(T) \prod_{i \in T} f(Y_i) \mu(Y_i) \cdot \prod_{i \in [k] \setminus T} \mu(Y_i)$$

Therefore,  $\sum_{y \in \mathcal{Y}} \Psi(y) \chi_S(y)$

$$\begin{aligned} &= \sum_{y \in \mathcal{Y}} \Psi(y) \prod_{i \in [k]} \chi_{S_i}(Y_i) \\ &= 2^k \sum_{y \in \mathcal{Y}} \left( \sum_{\substack{T \subseteq [k], \\ |T| \geq D}} \hat{\Gamma}(T) \prod_{i \in T} f(Y_i) \mu(Y_i) \cdot \prod_{i \in [k] \setminus T} \mu(Y_i) \right) \\ &\quad \prod_{i \in [k]} \chi_{S_i}(Y_i) \\ &= 2^k \sum_{\substack{T \subseteq [k], \\ |T| \geq D}} \hat{\Gamma}(T) \\ &\quad \sum_{y \in \mathcal{Y}} \left( \prod_{i \in T} f(Y_i) \mu(Y_i) \cdot \prod_{i \in [k] \setminus T} \mu(Y_i) \prod_{i \in [k]} \chi_{S_i}(Y_i) \right) \\ &= 2^k \sum_{\substack{T \subseteq [k], \\ |T| \geq D}} \hat{\Gamma}(T) \\ &\quad \sum_{Y_1, \dots, Y_k \in H} \left( \prod_{i \in T} f(Y_i) \mu(Y_i) \chi_{S_i}(Y_i) \right. \\ &\quad \left. \cdot \prod_{i \in [k] \setminus T} \mu(Y_i) \chi_{S_i}(Y_i) \right). \end{aligned}$$

Rearranging, we have  $\sum_{y \in \mathcal{Y}} \Psi(y) \chi_S(y) =$

$$\begin{aligned} &2^k \sum_{\substack{T \subseteq [k], \\ |T| \geq D}} \hat{\Gamma}(T) \prod_{i \in T} \left( \sum_{Y_i \in H} f(Y_i) \mu(Y_i) \chi_{S_i}(Y_i) \right) \\ &\quad \prod_{i \in [k] \setminus T} \left( \sum_{Y_i \in H} \mu(Y_i) \chi_{S_i}(Y_i) \right). \quad (7) \end{aligned}$$

Now, we will bound each product term in the outer sum by  $w^{-D/2}$ . We first observe that for every  $i \in [k]$ ,

$$\sum_{x \in H} \mu(x) \chi_{S_i}(x) \leq \sum_{x \in H} \mu(x) = 1.$$

If  $|S_i| \leq t$ , by (3)

$$\left| \sum_{x \in H} f(x) \mu(x) \chi_{S_i}(x) \right| \leq \frac{1}{w}.$$

If  $|S_i| > t$ , then

$$\left| \sum_{x \in H} f(x) \mu(x) \chi_{S_i}(x) \right| \leq \sum_{x \in H} \mu(x) = 1.$$

Since  $\sum_{i=1}^k |S_i| \leq tD/2$ , it follows that  $|S_i| \leq t$  for more than  $k - D/2$  indices  $i \in [k]$ . Thus, for each  $T \subseteq [k]$  such that  $|T| \geq D$ , there are at least  $D/2$

indices  $i \in T$  such that  $|S_i| \leq t$ . Hence,

$$\left| \sum_{y \in \mathcal{Y}} \Psi(y) \chi_S(y) \right| \leq 2^k w^{-\frac{D}{2}} \sum_{\substack{T \subseteq [k] \\ |T| \geq D}} |\hat{\Gamma}(T)| \leq 2^k w^{-\frac{D}{2}}.$$

Here, the last inequality is because  $|\hat{\Gamma}(T)| \leq 2^{-k}$  from (4).

From 1, 2 and 3, we have

$$\sum_{y \in \mathcal{Y}} \Psi(y) G(y) - W \max_{S \subseteq [d], |S| \leq \frac{tD}{2}} \left| \sum_{y \in \mathcal{Y}} \Psi(y) \chi_S(y) \right| > \gamma$$

if  $W \leq \gamma 2^{-k} w^{D/2}$ .  $\square$

We now derive Corollary 5.2. We need the following theorems on the approximate degree and the non-constant margin weight of the  $\text{OR}_d$  function.

**THEOREM 5.6** (APPROXIMATE DEGREE OF  $\text{OR}_d$  [27]).

$$\text{deg}_{1/3}(\text{OR}_d) = \Theta(\sqrt{d}).$$

**LEMMA 5.7** (MARGIN WEIGHT OF  $\text{OR}_d$ ).

$$w^*(\text{OR}_d|_{H_{d,1}}, t) \geq d/t.$$

**PROOF.** The function

$$\mu(x) = \begin{cases} 1/2 & \text{if } x = (1, \dots, 1), \\ 1/2d & \text{if } x \in H_{d,1} \setminus \{(1, \dots, 1)\}. \end{cases}$$

acts as the dual witness in Theorem 5.5.  $\square$

**PROOF OF COROLLARY 5.2.** We use Theorem 5.1 in the following setting. Let  $Y = \{-1, 1\}^{d/k}$ , the inner function  $f : Y \rightarrow \{-1, 1\}$  be  $\text{OR}_{d/k}$  and the outer function  $F : \{-1, 1\}^k \rightarrow \{-1, 1\}$  be  $\text{OR}_k$ ,  $\mathcal{Y} = H_{d,k}$  and  $H = H_{d/k,1}$ . By a simple counting argument, if  $(Y_1, \dots, Y_k) \in \{-1, 1\}^d \setminus H_{d,k}$ , then there exists  $i \in [k]$  such that  $Y_i \in \{-1, 1\}^{d/k} \setminus H_{d/k,1}$ . Further, by Theorem 5.6, we know that  $\text{deg}_{1/3}(F) = \Theta(\sqrt{k})$  and by Claim 5.7, we know that  $w^*(f|_H, t) \geq d/kt$ . Therefore, by Theorem 5.1, we have that, for every  $t \in \mathbb{Z}_+$ ,

$$\text{deg}_{1/6, W}(\text{OR}_d|_{H_{d,k}}) = \Omega\left(t\sqrt{k}\right)$$

for every

$$W \leq \frac{1}{6} 2^{-k} \left(\frac{d}{kt}\right)^{C\sqrt{k}}.$$

We obtain the conclusion by taking

$$t = \lfloor (d/k)(6W2^k)^{-1/C\sqrt{k}} \rfloor.$$

Since  $W \leq (1/6)2^{-k}(d/k)^{C\sqrt{k}}$ , it follows that  $t \geq 1$  and hence is a valid choice for  $t$  in applying Theorem 5.1.  $\square$

**Comparison to [29].** As described in the beginning of Section 4, Lemma 8 of the work of Servedio et al. [29] can be shown to imply that any polynomial  $p$  of weight  $W$  that  $1/3$ -approximates the  $\text{OR}_d$  function on all Boolean inputs requires degree  $\Omega(d/\log W)$ .<sup>1</sup> The proof in [29] relies on a

<sup>1</sup>More precisely, [29, Lemma 8] as stated shows that if the coefficients of a univariate polynomial  $P$  each have absolute

Markov-type inequality that bounds the derivative of a univariate polynomial in terms of its degree and the size of its coefficients. The proof of this Markov-type inequality is non-constructive and relies on complex analysis.

Here, we observe that our dual witness construction used to prove Corollary 5.2 also yields a general lower bound on the tradeoffs achievable between the weight and degree of the approximating polynomial  $p$ , even when we require  $p$  to be accurate only on inputs of Hamming weight at most  $O(\log W)$  (see Theorem 5.8). The methods of Servedio et al. do not yield any non-trivial lower bound on the degree in this setting. We also believe our proof technique is of interest in comparison to the methods of Servedio et al. as it is constructive (exhibiting an explicit dual witness for the lower bound) and avoids the use of complex analysis.

**THEOREM 5.8.** *Any polynomial  $p$  of weight  $W$  that  $1/6$ -approximates the  $\text{OR}_d$  function on all Boolean inputs requires degree  $d/2^{O(\sqrt{\log W})}$ .*

**PROOF.** As  $p$  is accurate on the entire Boolean hypercube, it is accurate on inputs of Hamming weight at most  $\log W$ . The theorem follows by setting  $k = \log W$  in the statement of Corollary 5.2.  $\square$

## 6. DISCUSSION

We gave a differentially private online algorithm for answering  $k$ -way marginal queries that runs in time  $2^{o(d)}$  per query, and guarantees accurate answers for databases of size  $\text{poly}(d, k)$ . More precisely, we showed that if there exists a polynomial of degree  $t$  and weight  $W$  approximating the  $d$ -variate OR function on Boolean inputs of Hamming weight at most  $k$ , then a variant of the private multiplicative weights algorithm can answer  $k$ -way marginal queries in time roughly  $\binom{d}{t}$  per query and guarantee accurate answers on databases of size roughly  $W\sqrt{d}$ . To this end, we gave a new construction showing the existence of polynomial approximations to the OR function on inputs of low Hamming weight. Specifically, we showed that polynomials of weight  $d^{0.01}$  and degree  $d^{1-\Omega(1/\sqrt{k})}$  exist.

**Practical Considerations.** Our algorithm for answering  $k$ -way marginals is essentially the same as in [19] except for using a different set of base functions (specifically, the set of all low-degree parities), which leads to an efficiency gain. We note that our algorithm degrades smoothly to the private multiplicative weights algorithm as the degree of the promised polynomial approximation increases, and never gives a worse running time. This behavior suggests that our algorithm may lead to practical improvements even for relatively small values of  $d$ , for which the asymptotic analysis does not apply. In such cases one might use an alternative

value at most  $W$ , and  $1/2 \leq \max_{x \in [0,1]} |P(x)| \leq R$ , then  $\max_{x \in [0,1]} |P'(x)| = O(\deg(P) \cdot R \cdot (\log W + \log \deg(P)))$ , where  $P'(x)$  denotes the derivative of  $P$  at  $x$ , and  $\deg(P)$  denotes the degree of  $P$ . By inspection of the proof, it is easily seen that if the  $L_1$ -norm of the coefficients of  $P$  is bounded by  $W$ , then the following slightly stronger conclusion holds:  $\max_{x \in [0,1]} |P'(x)| = O(\deg(P) \cdot R \cdot \log W)$ . When combined with the symmetrization argument of [29], this stronger conclusion implies: any polynomial  $p$  of weight  $W$  that  $1/3$ -approximates the  $\text{OR}_d$  function on all Boolean inputs requires degree  $\Omega(d/\log W)$ .

but similar analysis that shows the existence of a polynomial of degree  $kd^{1-c/k}$  and weight  $d^c$  (for any  $0 < c < k$ ) that *exactly* computes the  $d$ -variate OR function on inputs of Hamming weight at most  $k$ . Such a polynomial may be obtained as in our construction, by breaking the  $d$ -variate OR function into an OR of ORs, and using a degree  $k$  polynomial defined via polynomial interpolation, instead of Chebyshev polynomials, to approximate the outer OR on inputs of Hamming weight at most  $k$ . This variant does achieve asymptotic properties not shared by the algorithm of Theorem 1.1, owing to the fact that it uses exact rather than approximate representations of the database: for any constant  $k$ , this variant also runs in time  $\exp(d^{1-\Omega(1)})$ , and achieves worst-case additive error  $o(1)$  for sufficiently large databases, i.e., for  $n \gtrsim \tilde{O}(d^c)$ .

**Relationship with [12, 26].** As we mentioned in the introduction, in subsequent work, Dwork et al. [12] show how to privately release marginals in a very different parameter regime from what we consider here. Although their algorithm is quite different from ours, there are some important similarities. Their algorithm is based on a recent algorithm of Nikolov et al. [26] for answering arbitrary counting queries. This algorithm proceeds as follows. First, it adds noise  $O(|Q|^{1/2}/|D|)$  to the answer to every query to obtain a vector of noisy answers. The noise is sufficient to ensure differential privacy. Second, it “projects” the vector of noisy answers onto the convex body  $K$  consisting of all vectors of answers that are consistent with some real database. Surprisingly, the projection step will improve the accuracy. In fact, Nikolov et al. [26] show that the projected answers will be accurate even for very small databases. However current best algorithms for projecting on the body  $K$  require time  $2^{O(d)}$  for an arbitrary set of queries.

Dwork et al. [12] show that for the set of  $k$ -way marginals there is a convex body  $L$  such that the projection into  $L$  can be computed in time  $\text{poly}\binom{d}{\leq k}$  and  $L$  approximates  $K$  well enough to achieve accuracy on smaller databases than would be achievable with independent noise (however, these databases still have size  $d^{\Omega(k)}$ ). Our approximation of  $f_D$  by low-weight polynomials of degree  $t = o(d)$  can be shown to imply a polytope  $L'$  with  $2^{o(d)}$  vertices—which is sufficient to imply that projection can be computed in time  $2^{o(d)}$ —that approximates  $K$  well enough to achieve accuracy on databases of nearly optimal size (i.e., size roughly  $kd^{\Theta(1)}$ ). Thus our approximation-theoretic approach is relevant for understanding the capabilities and limitations of algorithms in the Nikolov et al. [26] framework.

**Future Directions.** Our lower bounds show that our polynomial approximation to the  $\text{OR}_d$  function on inputs of Hamming weight  $k$  is essentially the best possible; in particular, we cannot hope to substantially improve the running time on  $\text{poly}(d, k)$  size databases by giving approximating polynomials with better weight and degree bounds. This also rules out several natural candidates that can themselves be computed exactly by a low-weight polynomial of low-degree (e.g., the set of small-width conjunctions).

However, we do not know if it is possible to do better by using different feature spaces (other than the set of all low-degree monomials) to approximate all disjunctions over  $d$  variables. There is some additional evidence from prior work

that low-degree monomials may be the optimal choice: if the parameter of significance is the *size* of the set of functions used to approximate disjunctions on inputs of Hamming weight at most  $k$ , then low-degree monomials are indeed optimal [31] (see also Section 5 in the full version of [34]). It would be interesting to determine whether this optimality still holds when we restrict the  $L_1$  weight of the linear combinations used in the approximations to be  $\text{poly}(d)$ .

**Acknowledgments.** We thank Salil Vadhan for helpful discussions about this work.

## 7. REFERENCES

- [1] 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 *PODS*, pages 273–282, 2007.
- [2] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the sulq framework. In *PODS*, pages 128–138, 2005.
- [3] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *STOC*, pages 609–618, 2008.
- [4] J. Bruck and R. Smolensky. Polynomial Threshold Functions,  $AC^0$  Functions, and Spectral Norms. *SIAM J. Comput.*, 21(1):33–42, 1992.
- [5] M. Bun and J. Thaler. Dual lower bounds for approximate degree and markov-bernstein inequalities. In *Proceedings of 40th International Colloquium on Automata, Languages, and Programming*, pages 303–314, 2013.
- [6] M. Bun, J. Ullman, and S. P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. *CoRR*, abs/1311.3158, 2013.
- [7] M. Cheraghchi, A. Klivans, P. Kothari, and H. K. Lee. Submodular functions are noise stable. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 1586–1592, 2012.
- [8] A. De. Lower bounds in differential privacy. In *TCC*, pages 321–338, 2012.
- [9] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *PODS*, pages 202–210, 2003.
- [10] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC '06*, pages 265–284, 2006.
- [11] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. P. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC '09*, pages 381–390, 2009.
- [12] C. Dwork, A. Nikolov, and K. Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. *CoRR*, abs/1308.1385, 2013.
- [13] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *CRYPTO*, pages 528–544, 2004.
- [14] C. Dwork, G. N. Rothblum, and S. P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60, 2010.
- [15] V. Feldman and P. Kothari. Learning coverage functions. *Manuscript*, 2013.

- [16] A. Gupta, M. Hardt, A. Roth, and J. Ullman. Privately releasing conjunctions and the statistical query barrier. In *STOC '11*, pages 803–812, 2011.
- [17] A. Gupta, A. Roth, and J. Ullman. Iterative constructions and private data release. In *TCC*, pages 339–356, 2012.
- [18] M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. *NIPS '12*, 2012.
- [19] M. Hardt and G. N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, pages 61–70, 2010.
- [20] M. Hardt, G. N. Rothblum, and R. A. Servedio. Private data release via learning thresholds. In *SODA*, pages 168–187, 2012.
- [21] P. Jain and A. Thakurta. Mirror descent based database privacy. In *APPROX-RANDOM*, pages 579–590, 2012.
- [22] S. P. Kasiviswanathan, M. Rudelson, A. Smith, and J. Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *STOC*, pages 775–784, 2010.
- [23] H. Klauck. On arthur merlin games in communication complexity. In *Proceedings of the 26th Annual Conference on Computational Complexity (CCC)*, CCC, pages 189–199, 2011.
- [24] A. R. Klivans and R. A. Servedio. Toward attribute efficient learning of decision lists and parities. In J. Shawe-Taylor and Y. Singer, editors, *COLT*, volume 3120 of *Lecture Notes in Computer Science*, pages 224–238. Springer, 2004.
- [25] P. M. Long and R. A. Servedio. Low-weight halfspaces for sparse boolean vectors. In *ITCS*, pages 21–36, 2013.
- [26] A. Nikolov, K. Talwar, and L. Zhang. The geometry of differential privacy: the sparse and approximate cases. In *STOC*, pages 351–360, 2013.
- [27] R. Paturi. On the degree of polynomials that approximate symmetric boolean functions. In *Proceedings of the twenty-fourth annual ACM Symposium on Theory of Computing*, pages 468–474, 1992.
- [28] A. Roth and T. Roughgarden. Interactive privacy via the median mechanism. In *STOC '10*, pages 765–774, 2010.
- [29] R. Servedio, L.-Y. Tan, and J. Thaler. Attribute-efficient learning and weight-degree tradeoffs for polynomial threshold functions. In *25th Annual Conference on Computational Learning Theory (COLT), JMLR Workshop and Conference Proceeding*, volume 23, pages 14.1–14.19, 2012.
- [30] A. A. Sherstov. The intersection of two halfspaces has high threshold degree. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, pages 343–362, 2009.
- [31] A. A. Sherstov. The pattern matrix method. *SIAM Journal on Computing*, 40(6):1969–2000, 2011.
- [32] A. A. Sherstov. The multiparty communication complexity of set disjointness. In *STOC*, pages 525–548, 2012.
- [33] A. A. Sherstov. Strong direct product theorems for quantum communication and query complexity. *SIAM J. Comput.*, 41(5):1122–1165, 2012.
- [34] J. Thaler, J. Ullman, and S. P. Vadhan. Faster algorithms for privately releasing marginals. In *ICALP*, pages 810–821, 2012.
- [35] J. Ullman. Answering  $n^{2+o(1)}$  counting queries with differential privacy is hard. In *STOC*, pages 361–370, 2013.
- [36] J. Ullman and S. P. Vadhan. PCPs and the hardness of generating private synthetic data. In *TCC '11*, pages 400–416, 2011.
- [37] R. Špalek. A dual polynomial for OR. *CoRR*, abs/0803.4516, 2008.