Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 126 (2015)

Memory, Communication, and Statistical Queries

Jacob Steinhardt* Gregory Valiant† Stefan Wager‡

March 22, 2016

Abstract

If a concept class can be represented with a certain amount of memory, can it be efficiently learned with the same amount of memory? What concepts can be efficiently learned by algorithms that extract only a few bits of information from each example? We introduce a formal framework for studying these questions, and investigate the relationship between the fundamental resources of memory or communication and the sample complexity of the learning task. We relate our memory-bounded and communication-bounded learning models to the well-studied statistical query model. This connection can be leveraged to obtain both upper and lower bounds: we show strong lower bounds on learning parity functions with bounded communication, as well as upper bounds on solving sparse linear regression problems with limited memory.

1 Introduction

The increasing scale of problems we want to solve has led to new computing architectures, ranging from GPUs to distributed networks of computers, for which performance is not constrained by the number of operations per second. Instead, the performance depends on resources such as the memory per processor or machine, or the amount of communication required by the algorithm. This trajectory of modern computing suggests that we revisit the theory of the learnable with an eye not only towards the usual resources of time (number of operations required) and data (number of examples required), but with a consideration of both memory and communication.

From a very different perspective, it seems both practically and philosophically important to understand what factors drive the difficulty of a learning problem. Intuitively, we might assume that easy problems can still be learned in the presence of various constraints, whereas hard problems have fragile solutions that cannot withstand them; under this view, robustness to resource constraints can provide us with a richer understanding of the hardness of learning than traditional PAC theory.[1] One natural question in this line is to explore which concept classes can be efficiently learned by an algorithm that does not require significantly more memory than is required to store the true concept—essentially by an algorithm that can learn the concept class "in memory" without requiring additional side computations or "scratch paper." Similarly, many seemingly easy learning tasks have the property that even if relatively little information from each example is used, learning is still possible and rapid; it is natural to ask for a characterization of the class of problems that

*jsteinhardt@cs.stanford.edu

†valiant@stanford.edu

‡swager@stanford.edu

[1] Beyond the memory and communication bounds studied in this paper, such constraints could also involve, e.g., limits on the class of algorithms that can be used or robustness requirements with respect to faulty hardware.

ISSN 1433-8092

can be learned in this way. While these constraints on memory or communication may seem stringent, it is tempting to argue that nearly all of the processes by which humans learn—in addition to requiring relatively few examples and time—have little memory overhead and allow for a considerable compression of the information given in each example (for instance, millions of pixels may simply be remembered as “a persimmon”).

1.1 Our Contributions

Motivated by the dual goals of attending to the memory and communication considerations of modern systems, and developing a more nuanced understanding of the difficulty of learning problems, we consider two settings for studying resource-constrained learning:

- **Multi-party communication:** m parties engage in an arbitrary multi-party, multi-round communication protocol; each party possesses a single labeled example, and is allowed to adaptively broadcast up to b bits in aggregate over the course of the entire protocol.
- **Memory-limited streaming:** examples are observed in a stream, and the learning algorithm has access to at most b bits of storage.

We now provide an informal summary of our main theorems, which connect each of the above settings to the statistical query model, in which the learner may query $\mathbb{E}_p[\psi(x)]$ for any function $\psi : \mathcal{X} \to [-1, 1]$, and receives the expectation up to some error τ. We then leverage these connections to obtain both upper and lower bounds for concrete learning problems.

Our first result shows a tight correspondence between concepts that are learnable via statistical queries, and those that are learnable using a small amount of communication per example:

Theorem. *A concept class is learnable from* $\operatorname{poly}(n)$ *parties with* $\mathcal{O}(\log n)$ *bits of communication per party if and only if it is learnable from* $\operatorname{poly}(n)$ *statistical queries of error* $1/\operatorname{poly}(n)$*.*

Our next theorem shows that classes of function that can be learned via statistical queries can also be learned with relatively little memory. We note that our proof of this result is of an existential nature, and the reduction may not be computationally efficient.

Theorem. *If a class of distributions* $\mathcal{F}$ *is learnable with* $\operatorname{poly}(n)$ *statistical queries of tolerance* $1/\operatorname{poly}(n)$*, then it is learnable with* $\operatorname{poly}(n, \log|\mathcal{F}|)$ *samples and* $\mathcal{O}(\log|\mathcal{F}|\log(n))$ *bits of storage.*

Communication Lower bounds

The equivalence between multi-party communication and statistical queries serves as a powerful tool allowing one to immediately translate exponential lower bounds for statistical queries into exponential lower bounds for bounded communication protocols. We illustrate this via lower bounds for the well studied problem of learning a parity (PARITY). Recall that the problem of learning a parity over n-bit examples is the problem of recovering an arbitrary set $S \subseteq \{1, \ldots, n\}$ given access to uniformly random examples $x \in \{0,1\}^n$ together with their label $\ell(x) = \sum_{i \in S} x_i \mod 2$.

Using the classical exponential SQ lower bounds for learning PARITY of Blum et al. [1994], as well as a slight strengthening of our multi-party communication result, we can show that any communication protocol for learning PARITY with at most $n/4$ bits communicated per party requires $2^{\Omega(n)}$ parties. This result differs in kind from previous communication lower bounds (e.g. Shamir [2014]), which all exhibit at most polynomial (and typically linear) trade-offs between the communication per party and the number of parties.

The following more striking corollary of our communication theorem is also established by using known statistical query lower bounds:

Corollary. *Suppose that each party receives $n/4$ uniformly random labeled examples from PARITY, and can communicate at most $b = n/4$ bits. Then, at least $2^{\Omega(n)}$ parties are needed to learn PARITY.*

This result is surprisingly strong: if each party is allowed to communicate arbitrarily, then with decent probability, 4 parties would suffice to learn the parity. Meanwhile, if each party is allowed to communicate $b = n$ bits, then $\mathcal{O}(n)$ parties would suffice to learn the parity; and if each party receives n examples rather than $n/4$ examples, then even if the communication per party is limited to $b = 1$ bit, $\mathcal{O}(n)$ parties would also be capable of learning the parity.

We also consider the special case of one-way communication, where each party can only communicate a single b-bit message. In this model, we find problems for which effectively no compression is possible. For example, we show that any algorithm for distinguishing parity with noisy labels from uniformly random labels using polynomially many examples requires at least $b = n - \mathcal{O}(\log n)$ bits of communication; note that setting $b = n$ would enable each party to just broadcast their full example. Our proof leverages connections between χ^2-divergence and statistical query dimension that lead to strong lower bounds via Assouad's method. The χ^2-divergence seems to in some sense be the "right" distance for analyzing communication- and memory-constrained problems, as we discuss in Section 4.

Upper Bounds

We also consider implications of our reduction from SQ to memory-constrained learning. In several common statistical learning problems, we seek to learn an n-dimensional parameter vector that only has $k \ll n$ non-zero entries. A natural question to ask is whether we need $\mathcal{O}(n)$ memory to solve such problems, or if maintaining memory that scales with k is enough. By first providing a sample-efficient statistical query algorithm for sparse linear regression, we obtain as a corollary of our memory theorem:

Corollary. *In n dimensions, k-sparse linear regression can be solved using $k \cdot \mathrm{polylog}(n)$ bits of storage and $n \cdot \mathrm{poly}(k)$ samples.*

The linear dependence on n is unavoidable, since Steinhardt and Duchi [2015] show the amount of memory times the number of samples must be at least $k \cdot n$. We note also that Steinhardt and Duchi [2015] provide a similar result with better polynomial dependence, but require strong assumptions on the covariance structure.

Open Problems and Follow-Ups

A natural question left open by the above results is whether there is any non-trivial separation between problems that can be solved with bounded memory and those that can be solved without any storage constraints. The problem of learning a parity is a natural candidate for such a separation: a brute force search can learn the parity using memory only $\mathcal{O}(n)$ and an exponential number of examples, whereas Gaussian elimination can solve parity with $\mathcal{O}(n)$ examples and n^2 bits of storage.

Conjecture 1. *For any constant $\epsilon > 0$, any algorithm for learning PARITY over uniformly random n-bit examples requires either at least $(\frac{1}{4} - \epsilon)n^2$ bits of storage or at least $2^{\Theta(n)}$ labeled examples.*[2]

[2] In an earlier version of this paper, ECCC 22:126 (2015), we stated this conjecture with $n^2/4$ bits of storage and $2^{n/4}$ samples; as pointed out by Raz [2016], that conjecture was too strong, and there is an algorithm that uses $2^{n/4}$ samples and solves PARITY using $9n^2/64 + o(n^2)$ bits of storage.

Since the dissemination of an earlier version of our paper, Ran Raz gave an extremely nice proof of this conjecture, with slightly weaker constants, showing that any algorithm for learning PARITY requires either $n^2/25$ bits of storage, or an exponential number of examples [Raz, 2016]. It would be very exciting if Raz' proof approach could be extended to the cell-probe model, where the algorithm is given access to read-only memory containing the examples. The analogous conjecture for this cell-probe model is that any algorithm either requires read/write memory $\Theta(n^2)$, or must make an exponential number of probes to the read-only memory containing the examples.

In a different direction, it would be useful to give a computationally efficient reduction from statistical query algorithms to memory-limited algorithms, as opposed to our implicit reduction. One more modest and practically relevant goal would be to provide an efficient memory-bounded algorithm for sparse regression without assumptions on the covariance structure.

1.2 Related Work

While there were a number of efforts to develop noise-tolerant learning algorithms, the introduction of the *statistical query* model of Kearns [1998] allowed for the development of a more general theory of how learning algorithms may be robust to noise, and the development of a common set of abstract tools for the design of such algorithms. Since the initial definition of the statistical query framework, there have been successful efforts to characterize which concepts can be learned within this framework [e.g., Blum et al., 1994], including recent results showing inherent connections between statistical query learnability and differential privacy [Gupta et al., 2011, Kasiviswanathan et al., 2011, Balcan and Feldman, 2013] and evolvability [Feldman, 2009].

Perhaps the line of investigation most closely related to ours is the recent work of Shamir [2014], which considers online learning given under memory and communication restrictions. The results of that paper show linear trade-offs between the amount of resources (memory, communication) and the number of samples required (i.e., halving the resources doubles the number of samples); in contrast, our communication lower bound yields an exponential tradeoff between sample complexity and communication (i.e., decreasing the resources by $\mathcal{O}(1)$ doubles the number of samples).

Several authors have also considered communication and/or privacy restrictions in the distributed data setting [Balcan et al., 2012, Duchi et al., 2013, Zhang et al., 2013, Garg et al., 2014, Braverman et al., 2015]. As in Shamir [2014], this work largely obtains only linear trade-offs between the information exchanged per sample/round and the sample complexity/number of rounds.

Ben-David and Dichterman [1998] consider several learning models in the streaming setting that restrict the information that may be extracted from each example, including the wRFA model, which is equivalent to the STAT-1 model of Feldman et al. [2013], as well as a restriction of our multi-party communication model (where each party must send all b bits at once). Ben-David and Dichterman derive a weaker version of our communication lower bound in this setting.

There is a huge literature on communication complexity (see Kushilevitz and Nisan [1997], Lee and Shraibman [2009] for surveys), and the related concept of information complexity from the theoretical computer science community [Chakrabarti et al., 2001, Bar-Yossef et al., 2004, Braverman and Rao, 2011]. Such work focuses on understanding the length or entropy/information content of messages that must be communicated between distributed parties in order to compute a desired function, such as the set intersection, of their individual inputs. Recently, it was shown that there exist problems that exhibit an exponential gap between the required message length and the required entropy of the communicated messages [Ganor et al., 2014].

Finally, there is a large body of work on memory bounded computation from the theory community [e.g., Reingold, 2008], and work on the applied side on reducing the memory needs [e.g., Langford et al., 2009, Xiao, 2010, Agarwal et al., 2012, Mitliagkas et al., 2013, Arora et al., 2013, Steinhardt et al., 2014] and/or communication needs [e.g., Niu et al., 2011] for various specific learning tasks, though this literature is beyond the scope of this paper. The topic of finite-memory learning has also received attention from the statistics community, with a focus on hypothesis testing [e.g., Cover, 1969, Hellman and Cover, 1970].

1.3 Preliminaries

We consider *learning problems* $(\mathcal{X}, \mathcal{F})$, where $\mathcal{F}$ is a *concept class* of distributions p on $\mathcal{X}$. Our goal is to learn $p \in \mathcal{F}$ given some indirect access to it (for instance, a sequence of i.i.d. samples). We say that an algorithm (ε, δ)-*learns* $\mathcal{F}$ if, for each $p \in \mathcal{F}$, with probability $1 - \delta$ the algorithm returns a $\hat{p}$ with $\rho(p, \hat{p}) \leq \varepsilon$ for some given distance function ρ. Unless otherwise specified, our results hold for arbitrary choice of ρ.

In the sequel, we will draw many connections with the *statistical query model.* In this model, rather than being given samples from $\mathcal{X}$, we are allowed to make queries to an "SQ oracle" $\mathrm{SQ}(\psi, \tau)$, which takes as input a function $\psi : \mathcal{X} \to [-1, 1]$ and a real number $\tau > 0$, and outputs a real number μ satisfying $|\mu - \mathbb{E}_{x \sim p}[\psi(x)]| \leq \tau$. Throughout, we will use the symbol m to denote the number of samples/queries/parties, and n to denote the ambient parameter controlling the hardness of the problem (for instance, the dimension or length of each example).

2 Multi-Party Communication

In this section, we consider the following multi-party communication model: each of m parties receives a sample from an unknown distribution $p \in \mathcal{F}$. The parties then communicate in a series of rounds, where in each round one of the parties broadcasts a single bit. The party to broadcast in a given round is determined (possibly stochastically) based on all of the other bits broadcast so far and its identity is common knowledge, and each party may communicate at most b bits in total. The final output concept $\hat{p}$ may depend on all of the communicated messages. An intermediate stage of this process is illustrated below, where c_j denotes the jth bit communicated by a given party:

c_3:			?		
c_2:	1		0		
c_1:	0		1	1	
party:	1	2	3	4	5

Here $b = 3$ and parties 1 and 3 have communicated 2 bits so far, while party 4 has communicated 1 bit so far; party 3 is about to communicate their third (and final) bit. A possible broadcasting order that could have produced the above figure is: party 1 communicates 0, party 4 communicates 1, party 1 communicates 1, party 3 communicates 1, party 3 communicates 0.

We are interested in what concept classes $\mathcal{F}$ can be learned given a polynomial number of total parties m. We will also at times consider the one-way communication model in which, if we number the parties $1, \ldots, n$, all parties with lower index must communicate before all parties with higher index. The primary results in this section connect the above multi-party communication model with the statistical query model. In particular, when $b = C\log(n)$, these two models are equivalent.

Theorem 2. *Any concept class $\mathcal{F}$ that can be (ε, δ)-learned from n parties, each given $C\log(n)$ bits of communication, can be $(\varepsilon, 2\delta)$-learned from $\mathrm{poly}(n)$ statistical queries of tolerance $\frac{\delta}{\mathrm{poly}(n)}$. Conversely, any concept class that can be (ε, δ)-learned from n statistical queries of tolerance $\frac{1}{n}$ can* *be* $(\varepsilon, 2\delta)$*-learned from* $\text{poly}(n, \log(1/\delta))$ *parties, each with* 1 *sample and* 1 *bit of communication, even in the one-way communication model.*

Note that the linear dependence on δ can be made logarithmic via standard amplification techniques.

The key idea is a reduction from communication-constrained algorithms to statistical query algorithms, which extends a similar reduction made by Ben-David and Dichterman [1998] for the "restricted focus of attention" model. We will describe the idea here, with a detailed exposition in Appendix A.1. Consider an intermediate state of the algorithm, where for instance a given party has already transmitted bits $c_1 = 1$, $c_2 = 0$, and is about to transmit bit c_3 (other parties could have made transmissions in the meantime, but these turn out to not matter for the proof). To simulate this transmission, it is enough to know the probability that c_3 is 1, where the probability is with respect to the party's hidden sample, and conditioned on the bits c_1 and c_2. We can calculate this as

$$\begin{aligned} p(c_3 = 1 \mid c_{1:2} = 10) &= p(c_{1:3} = 101)/p(c_{1:2} = 10) && (1) \\ &= \underbrace{\mathbb{E}[\mathbb{I}[c_{1:3} = 101]]}_{\text{statistical query}}/p(c_{1:2} = 10). && (2) \end{aligned}$$

We can calculate the indicated expectation using the statistical query $\psi(x) = \mathbb{I}[c_{1:3}(x) = 101]$. Note that if the statistical query has error τ, then the error in our conditional probability is $\tau/p(c_{1:2} = 10)$. The expected error (marginalizing over $c_{1:2}$) is then

$$\tau \cdot \mathbb{E}_{c_{1:2}}[1/p(c_{1:2})] = \tau \cdot \sum_{c_{1:2}} p(c_{1:2}) \cdot 1/p(c_{1:2}) = 4\tau; \tag{3}$$

in general, if j bits have been communicated so far, the expected error is $2^j\tau$. A rigorous version of this argument yields the following result.

Proposition 3. *If a concept class* $\mathcal{F}$ *can be* (ϵ, δ)*-learned with* m *parties and* b *bits of communication per party, then* $\mathcal{F}$ *can also be* $(\epsilon, 2\delta)$*-learned by a statistical query algorithm that makes* $2bm$ *statistical queries of tolerance* $\tau = \delta/\left(2^{b+1}m\right)$.

Parity lower bounds.

Proposition 3 yields strong lower bounds on the learnability of parity functions through multi-party communication. We recall the following bound of Blum et al. [1994] on the learnability of parity functions in the statistical query model:[3]

> Let $\mathcal{P}_{n,r}$ be the class of n-bit parity functions that depend on at most r coordinates: that is, $\mathcal{X} = \{0,1\}^n$, and $f_v(x) = (-1)^{\sum_{i=1}^n v_i x_i}$ for some $v \in \{0,1\}^n$ with $\|v\|_0 \leq r$. Also let $s = |\mathcal{P}_{n,r}|$. Then, for $s \geq 16$, no algorithm can $(1/3, 1/3)$-learn $\mathcal{P}_{n,r}$ with $s^{1/3}/2$ statistical queries of tolerance $s^{-1/3}$.

To bring this result in line with our notation, our concept class consists of the distributions $p_v(x, y) = \frac{1}{2^n}\mathbb{I}[y = f_v(x)]$, where $(x, y) \in \{0,1\}^n \times \{0,1\}$, and the distance measure $\rho(p_v, p_{v'})$ is $\mathbb{P}[f_v(x) \neq f_{v'}(x)]$. As a consequence of the above result, it is impossible to learn the class of parity functions in the multi-party communication setting with less than $n/4$ bits per party:

[3] The theorem as stated in Blum et al. [1994] is for $(1/3, 0)$-learning, but the same proof establishes the impossibility of $(1/3, 1/3)$-learning as well.

Corollary 4. *Given b bits of communication per party, no algorithm can $(1/3, 1/6)$-learn the class $\mathcal{P}_{n,r}$ with less than $|\mathcal{P}_{n,r}|^{1/3} / \left(2^{b+4}\right)$ parties. In particular, no algorithm can $(1/3, 1/6)$-learn $\mathcal{P}_{n,n}$ with less than $2^{n/3-b-4}$ parties, and hence any algorithm communicating at most $n/4$ bits per party requires an exponential number of parties.*

We can further strengthen Corollary 4 by adapting a result from Blum et al. [2003] concerning "k-ary" statistical queries, which are statistical queries that can depend on k independent examples rather than only a single example. Roughly, Blum et al. [2003] show that using k-ary queries instead of unary queries can only improve the query complexity by a factor of $2^{\mathcal{O}(k)}$. In the concrete case of learning parity functions, we use this to show that PARITY is hard in the multi-party setting even if each party is given $\Omega(n)$ examples at once:

Corollary 5. *Given b bits of communication and k i.i.d. examples per party, no algorithm can $(1/3, 1/32)$-learn $\mathcal{P}_{n,n}$ with less than $2^{(n-k-2b)/3-7}/(bn)$ parties. In particular, any algorithm using $\frac{n}{4}$ bits and $\frac{n}{4}$ examples per party requires $2^{\Omega(n)}$ parties.*

Statistical queries to one-way communication.

Finally, we turn to the other direction of Theorem 2, showing that any statistical query algorithm can be simulated with one-way communication. This direction is straightforward: to simulate a statistical query (ψ, τ), we draw $1/\tau^2$ samples $x \sim p$ and communicate $\psi(x)$ for each sample. In fact, it suffices to communicate an unbiased estimate of $\psi(x)$, by sending $+1$ with probability $\frac{1+\psi(x)}{2}$ and -1 with probability $\frac{1-\psi(x)}{2}$. Since the statistical queries are sequential, information need only be sent in one direction. This yields Proposition 6 below, which combined with Proposition 3 proves Theorem 2.

Proposition 6. *If a concept class $\mathcal{F}$ is (ε, δ)-learnable with m statistical queries of tolerance τ, then it is $(\varepsilon, 2\delta)$-learnable with $2m \log(2m/\delta)/\tau^2$ samples and 1 bit of communication per sample.*

3 Memory-Limited Streaming

In this section, we will consider a streaming model: in each round i, the algorithm observes a sample $X^{(i)} \sim p$, and updates its state from $Z^{(i-1)}$ to $Z^{(i)}$, where $Z^{(i)} \in \{0,1\}^b$ may only depend on $X^{(i)}$ and $Z^{(i-1)}$. The output concept $\hat{p}$ may only depend on the final memory state $Z^{(m)}$:

$$\begin{tikzcd}
 & p \arrow[dl] \arrow[d] \arrow[drr] & & & \\
X^{(1)} \arrow[d] & X^{(2)} \arrow[d] & \cdots & X^{(m)} \arrow[d] & \\
Z^{(1)} \arrow[r] & Z^{(2)} \arrow[r] & \cdots \arrow[r] & Z^{(m)} \arrow[r] & \hat{p}
\end{tikzcd}$$

Note that any learning algorithm that eventually achieves zero error must have $b \geq \log|\mathcal{F}|$. We are interested in which concept classes admit learning algorithms that approach this threshold (i.e., $b = \log|\mathcal{F}| \operatorname{polylog}(n)$).

As before, we will draw a connection with the statistical query model; any concept class which is learnable from statistical queries can be learned with nearly optimal memory:

Theorem 7. *If a class $\mathcal{F}$ is $(\varepsilon, 0)$-learnable with m_0 statistical queries of tolerance τ, then for any k it is (ε, δ)-learnable with at most*

$$m = \mathcal{O}\left(\frac{\lceil m_0/k \rceil \log |\mathcal{F}|}{\tau^2} \left(\log\log|\mathcal{F}| + \log(m_0) + \log(1/\delta)\right)\right) \text{ samples and} \tag{4}$$

$$b = \mathcal{O}\left(\log|\mathcal{F}| \left(\log(m_0) + \log\log(1/\tau)\right) + k\log(1/\tau)\right) \text{ bits of memory.} \tag{5}$$

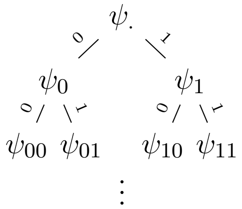
Setting $k = 1$, we see that any class that is learnable with n statistical queries of tolerance $\frac{1}{n}$ is learnable from a stream of $\mathrm{poly}(n, \log|\mathcal{F}|)$ examples and $b = \mathcal{O}(\log|\mathcal{F}| \log(n))$ bits of memory. Since we allow the SQ algorithm to be an improper learner, we can extend Theorem 7 to the setting where $|\mathcal{F}| = \infty$ and we instead consider an ε-covering of $|\mathcal{F}|$ under ρ (in this case we need ρ to be a metric). For simplicity, however, we focus on the case of finite $\mathcal{F}$.

We also note the roughly linear trade-off between sample complexity and memory as we vary k; this behavior parallels other known results on memory-limited learning [Shamir, 2014, Steinhardt and Duchi, 2015], suggesting that Theorem 7 is close to the "right" answer. However, there are also at least some problems that can be learned with limited memory but not statistical queries, such as parity on $\sqrt{n}$ bits.

Proof Sketch for Theorem 7

The naïve approach of simply remembering the results of the statistical queries will not work, since the number m of statistical queries could be much larger than $\log|\mathcal{F}|$. Instead, we show that it is always possible to identify a subset of $\mathcal{O}(\log|\mathcal{F}|)$ "important" queries, such that remembering the results of only these queries suffices to recover the answer.

To do this, we first binarize the output of each statistical query $\mu = \mathrm{SQ}(\psi, \tau)$ by reporting 1 if $\mu > t$ and 0 if $\mu \le t$ for a specified threshold t. We can then think of an SQ algorithm as following a decision tree based on the output of each binarized query:



Call an edge *light* if it eliminates at least half of the remaining concepts, and *heavy* otherwise. Note that every path has at most $\log_2|\mathcal{F}|$ light edges. As long as not both of the children are heavy, we can uniquely remember a path by keeping track of only the light edges, leading to memory usage $\mathcal{O}(\log|\mathcal{F}|\log(n))$.

Since the queries have error τ, it is possible for both edges from a node to be heavy if many concepts satisfy $\mathbb{E}_p[\psi(x)] \in [t - \tau, t + \tau]$. However, we can always avoid such cases: for a given query $\mathrm{SQ}(\psi, \tau/2)$, the concepts compatible with a 0 response at threshold $t - \tau/2$ and a 1 response at threshold $t + \tau/2$ are disjoint, so at least one of these two binarized queries has a light edge; we can always replace a query $\mathrm{SQ}(\psi, \tau)$ at threshold t with either of the two queries above while obtaining strictly more information, so we can always refine a query to have at least one light edge.

The above argument establishes Theorem 7 in the case $k = 1$. To extend to general k, we compute all of the next k upcoming queries in parallel, under the assumption that all of the next steps in the algorithm will follow the heavy edge. If our assumption is correct, then we successfully skip k steps down the tree, and if it is incorrect, we eliminate at least half of the concepts; in either case we make substantial progress. We next show how Theorem 7 applies to a concrete learning problem.

Application: Sparse Linear Regression

Theorem 7 yields a low-memory algorithm for sparse linear regression. We assume we are given a stream of samples $(x^{(i)}, y^{(i)}) \in [-1, 1]^n \times [-\mathcal{O}(k), \mathcal{O}(k)]$, where the x are drawn from a known distribution $x \sim \mathcal{D}$, and $y = w^* \cdot x + v$, where v is drawn from a known distribution $\mathcal{D}_v$ with zero mean independent of x. We also assume that $\|w^*\|_0 \leq k$ and $\|w\|_1 \leq \mathcal{O}(k)$ for some $k \ll n$. Letting $f(w) = \mathbb{E}_{x\sim\mathcal{D}}[(y - w \cdot x)^2]$, our goal is to recover a vector $\hat{w}$ such that $f(\hat{w}) - f(w^*) \leq \varepsilon$. Therefore, in this case $\rho(w, w') = \mathbb{E}[((w' - w) \cdot x + v)^2]$, which yields $\rho(w, w^*) = f(\hat{w}) - f(w^*)$.

The typical approach to solving such regression problems in the streaming setting is to use a stochastic optimization algorithm such as exponentiated gradient [Kivinen and Warmuth, 1997], which can learn w^* up to error ε in $\text{poly}(k, \log(n), \varepsilon^{-1})$ samples. The key observation is that each step of the exponentiated gradient algorithm consists of making n statistical queries, and furthermore if each query has error τ then the resulting increase in $f(\hat{w})$ is at most $\mathcal{O}\left(k\tau^2\right)$ (c.f. Lemma 21 in the appendix). As a consequence, we have:

Theorem 8. *The k-sparse linear regression problem described above can be (ε, δ)-learned using $\mathcal{O}\left(k \log^2\left(\frac{n}{\varepsilon}\right)\right)$ bits of memory and $\widetilde{\mathcal{O}}\left(\frac{nk^8 \log \delta^{-1}}{\varepsilon^4}\right)$ samples.*

We remark that simply running the exponentiated gradient algorithm would require $\Omega(n)$ memory. The only other result we are aware of that efficiently solves sparse linear regression with sublinear memory is the algorithm of Steinhardt and Duchi [2015], which works even in the agnostic setting but requires an incoherence assumption on X that we avoid. Their algorithm requires only $\widetilde{\mathcal{O}}\left(kd/b\right)$ samples given b bits of memory, suggesting that we could improve the polynomial factors in our algorithm; they also establish a lower bound of $\Omega(kd/b)$—the dependence on n therefore cannot be removed without further assumptions.

Intriguingly, we can identify at least one assumption under which the above lower bound can be surpassed—namely, if we assume that the covariates x are r-sparse for some $r \ll n$. Then, we can again run exponentiated gradient, and use the count-min sketch algorithm of Cormode and Muthukrishnan [2005] to estimate the gradient using only a small number of statistical queries in each iteration. This yields:

Theorem 9. *Under the conditions of Theorem 8, suppose in addition that $\|x\|_1 \leq r$ for all x. Then, the k-sparse linear regression problem can be (ε, δ)-learned with $\mathcal{O}\left(k \log^2\left(n/\varepsilon\right)\right)$ bits of memory and $\widetilde{\mathcal{O}}\left(r^3 k^{10} \log^2(1/\delta)/\varepsilon^5\right)$ samples.*

4 χ^2-divergence, Streaming Algorithms, and SQ Dimension

In this final section, we present analysis tools that yield particularly sharp lower bounds in the one-way communication setting. These tools draw connections between the statistical query dimension, χ^2-divergence, and Assouad's method. Surprisingly, for communication- and memory- constrained problems, the χ^2-divergence appears to be a more natural distance than the more common KL divergence, for reasons we will explain at the end of the section.

As motivation for our analysis, we consider the problem of distinguishing a noisy parity (in which labels are corrupted to be uniformly random with probability ε) from examples where the labels are completely random. In the interest of generality, we assume that each party receives k

training examples, which we call a "k-ary example", and must then compress these examples into a single b-bit message. We are able to show a nearly tight communication lower bound for this problem:

Proposition 10. *Given* $\mathrm{poly}(n)$ *k-ary examples and* $n - k - \omega(\log n)$ *bits of communication, no one-way communication algorithm can distinguish a noisy parity (with fixed noise level ε) from uniformly random bits with probability greater than $\frac{3}{4}$.*

Note that we can solve the above problem with $n-k+1$ bits and $\mathcal{O}(n)$ k-ary examples: simply find a linear combination of the k examples whose first k bits are zero, and communicate the remaining bits and corresponding label.

Proposition 10 is based on a more general result, which will require some notation to state. First, given $p, p' \in \mathcal{F}$, and a base distribution p_0, define the χ^2*-product*

$$[p, p'] \stackrel{\text{def}}{=} \sum_{x \in \mathcal{X}} \frac{p(x)p'(x)}{p_0(x)} - 1. \tag{6}$$

This product is closely related to the χ^2-divergence defined by $D_{\chi^2}(p\|q) = \sum_{x\in\mathcal{X}} q(x)^2/p(x) - 1$. Indeed, $[p,p]$ is simply $D_{\chi^2}(p_0\|p)$, while

$$[p, p'] = 2D_{\chi^2}\left(p_0 \middle\| \frac{p+p'}{2}\right) - \frac{1}{2}\left(D_{\chi^2}(p_0\|p) + D_{\chi^2}(p_0\|p')\right). \tag{7}$$

For any functions $p_1, \dots, p_n \in \mathcal{F}$, further define the $n \times n$ matrix

$$M_{ij} \stackrel{\text{def}}{=} [p_i, p_j]. \tag{8}$$

As an example, suppose that $\mathcal{F}$ defines a binary labeling task, so that $\mathcal{X} = \mathcal{X}_0 \times \{-1, +1\}$, and $p_i(x_0, y) = p(x_0)\mathbb{I}[f_i(x_0) = y]$ for some labeling function f_i. If we further define the base distribution $p_0(x_0, y) = p(x_0)/2$, we can check that $M_{ii} = 1$ while $M_{ij} = \mathbb{P}[f_i(x_0) = f_j(x_0)] - \mathbb{P}[f_i(x_0) \neq f_j(x_0)]$.

We will find that the difficulty of learning with constrained one-way communication depends on the largest eigenvalue of the matrix M, i.e., it depends on $\lambda = \lambda_{\max}(M)$. The following result strengthens the SQ-based lower bound (Proposition 3) in the one-way communication setting.

Theorem 11. *For any distributions $p_1, \dots, p_n \in \mathcal{F}$, let p_0 be a base distribution such that $\varepsilon p_0 \leq p_i$ for each i. Then, assuming that p_i is drawn uniformly at random from $\{p_1, \dots, p_n\}$, any one-way communication algorithm requires at least $\frac{n\varepsilon}{2^{b+2}((\lambda+1)^k - 1)}$ k-ary examples to distinguish p_i from p_0 with probability greater than $\frac{3}{4}$.*

For the case of learning parities, there are in fact $2^n - 1$ distributions such that $[p_i, p_j] = 0$ for all $i \neq j$, in which case $\lambda_{\max}(M) = 1$; this yields Proposition 10.

The constraint $p_i \geq \varepsilon p_0$ can be interpreted as saying that examples are corrupted with noise from p_0 with probability ε. If our algorithm can tolerate noise (for instance, if it is based on statistical queries), we can always define $\tilde{p}_i = (p_i + p_0)/2$, in which case $\varepsilon = \frac{1}{2}$. Even if our algorithm cannot tolerate noise, we can set $\tilde{p}_i = (1-\varepsilon)p_i + \varepsilon p_0$ for $\varepsilon = 1/8n$, in which case with probability $\frac{7}{8}$ the noise will not manifest even once over the course of the algorithm. This allows us to extend Theorem 11 to the noiseless case, with a lower bound that is weaker by a square-root.

The matrix M that underlies Theorem 11 is closely related to the statistical query dimension. In terms of our notation, and again restricting to binary labeling tasks, the statistical query dimension of $\mathcal{F}$ is the largest n such that there is an $n \times n$ matrix M with $|M_{ij}| \leq \frac{1}{n}$ for all $i \neq j$ [Blum et al., 1994]. Such a matrix M can have a maximum eigenvalue of at most 2, thus immediately yielding the following corollary:

Corollary 12. *If $\mathcal{F}$ is a labeling problem with statistical query dimension n, then any one-way communication algorithm for learning $\mathcal{F}$ requires at least $\frac{n\varepsilon}{2^{b+2}(3^k-1)}$ samples to distinguish p_i from p_0 with probability greater than $\frac{3}{4}$.*

Since statistical query algorithms can be simulated with 1 bit of communication (c.f. Proposition 6), this result recovers classical SQ lower bounds [Blum et al., 1994, Szörényi, 2009] up to logarithmic factors.

Proving Theorem 11

Theorem 11 is based on *Assouad's method* [Assouad, 1983], a lower bound technique from the information theory literature which has recently found success in analyzing adaptive communication algorithms [Arias-Castro et al., 2013, Duchi et al., 2013, Steinhardt and Duchi, 2015]. We re-state a variant of the key lemma in Assouad's method below; here $p_1, \dots, p_n$ are possible distributions from which the data could be drawn.

Lemma 13. *Let $Z^{(1:m)}$ be the messages sent by a one-way communication algorithm, and suppose that we have the following bound on the cumulative one-step χ^2-divergence:*

$$\frac{1}{n}\sum_{j=1}^{m}\sum_{i=1}^{n}\mathbb{E}_{z^{(1:j-1)}\sim p_0}\left[D_{\chi^2}\left(p_i(Z^{(i)} \mid z^{(1:i-1)})\middle\|p_0(Z^{(i)} \mid z^{(1:i-1)})\right)\right] \leq D. \tag{9}$$

Then, if we are given samples from either p_0 (with probability $1/2$) or a uniformly chosen p_i, the probability of distinguishing between these two cases is at most $\frac{1}{2}+\sqrt{D/8}$.

Our main innovation is a lemma bounding the χ^2-divergence in Lemma 13 in terms of λ:

Lemma 14. *For any $z = z^{(j)}$ and $\hat{z} = z^{(1:j-1)}$, we have*

$$\sum_{i=1}^{n}(p_i(z \mid \hat{z}) - p_0(z \mid \hat{z}))^2 \leq \lambda_{\max}(M)p_0(z \mid \hat{z}). \tag{10}$$

Noting that $D_{\chi^2}(p\|q)$ can alternately be expressed as $\sum_{x\in\mathcal{X}}\frac{(q(x)-p(x))^2}{p(x)}$, we see that Lemma 14 implies a bound on the χ^2-divergence of

$$\sum_{i=1}^{n}D_{\chi^2}\left(p_i(Z \mid \hat{z})\|p_0(Z \mid \hat{z})\right) = \sum_{z\in\{0,1\}^b}\sum_{i=1}^{n}\frac{(p_i(z \mid \hat{z}) - p_0(z \mid \hat{z}))^2}{p_i(z \mid \hat{z})} \tag{11}$$

$$\leq \lambda_{\max}(M)\sum_{z\in\{0,1\}^b}\frac{p_0(z \mid \hat{z})}{p_i(z \mid \hat{z})} \leq \lambda_{\max}(M)\sum_{z\in\{0,1\}^b}\frac{1}{\varepsilon} = 2^b\lambda_{\max}(M)/\varepsilon. \tag{12}$$

Note how this bound takes the memory constraint into account: as long as we can pointwise bound the sum over z, we get a bound that depends on 2^b (the number of distinct z's). The preceding bound together with Lemma 13 yields Theorem 11 in the case that $k = 1$. For the general case where individual parties observe groups of k examples, we use the following "tensorization lemma".

Lemma 15. *Let $p_i^{\otimes k}$ denote the distribution on $\mathcal{X}^k$ consisting of k independent samples from p_i, and define $M^{(k)}$ by $M_{ij}^{(k)} = [p_i^{\otimes k}, p_j^{\otimes k}]$, where the base distribution is $p_0^{\otimes k}$. Then, for each i, j, we have $M_{ij}^{(k)} = (M_{ij}+1)^k - 1$. In particular, $\lambda_{\max}(M^{(k)}) \leq (\lambda_{\max}(M)+1)^k - 1$.*

Discussion: χ^2- vs. KL-divergence.

The KL-divergence is a versatile measure of distance between two distributions. However, a growing body of work suggests that, for studying learning with memory and communication constraints, the χ^2-divergence is more natural [Shamir, 2014, Steinhardt and Duchi, 2015]. We will give some intuition here for why this might be the case.

Ignore memory constraints for now, and suppose that we observe $4n$ examples for the parity problem with noise $\varepsilon = \frac{1}{2}$. We consider two possible actions: (i) store all $4n$ examples with probability $\frac{1}{4n}$, or (ii) store a single example. It is clear that these two situations are very different—in (i), we have (with probability close to $\frac{1}{4n}$) enough information to solve the problem, whereas in (ii) we have barely made any progress. However, the expected KL distance in both cases is the same, namely $\frac{1}{2}\log(4/3)$; c.f. Lemma 22. On the other hand, the expected χ^2 distances are very different: $\frac{1}{4n}\left(\left(\frac{4}{3}\right)^{4n} - 1\right)$ in the first case and $\frac{1}{3}$ in the second case. This accurately reflects the much larger amount of progress toward recovering the parity made in the first case.

Generalizing this idea, we prove the following results showing that learning with only information-theoretic constraints on communication or memory is not hard. There thus appears to be a fundamental difference between information-theoretic bits and "physical" bits when considering the difficulty of learning. Because the χ^2-divergence is sensitive to this difference, it appears to be a better analysis tool.

First, we can show that, if our only constraint is on the information-theoretic entropy of the memory state (rather than the number of physical bits of memory), then there is essentially no separation between PAC-learnability and memory-constrained PAC-learnability:

Proposition 16. *If ρ is a metric, then any concept class that is (ε, δ)-learnable with* $\mathrm{poly}(n)$ *memory and samples can be $(2\varepsilon, \delta)$-learned from* $\mathrm{poly}(n)$ *samples with a memory state whose entropy never exceeds* $\log_2 |\mathcal{F}| + \mathcal{O}(\log n)$ *bits.*

In Proposition 16, we use the fact that ρ is a metric to reduce from improper learning to proper learning (by replacing an output concept $\hat{p} \notin \mathcal{F}$ by some nearby concept in $\mathcal{F}$); note that this reduction is not computationally efficient in general.

Similarly to Proposition 16, if our constraint is on the number of information-theoretic bits of communication, then there is essentially no separation between PAC-learnability and communication-constrained PAC-learnability:

Proposition 17. *If a concept class is (ε, δ)-learnable from* $\mathrm{poly}(n)$ *samples that can each be represented with* $\mathrm{poly}(n)$ *bits, then it is (ε, δ)-learnable from* $\mathrm{poly}(n)$ *samples and one-way communication with messages each of whose entropies are at most* 1 *bit.*

References

A. Agarwal, S. Negahban, and M.J. Wainwright. Stochastic optimization and sparse statistical recovery: Optimal algorithms for high dimensions. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

E. Arias-Castro, E.J. Candes, and M.A. Davenport. On the fundamental limits of adaptive sensing. *IEEE Transactions on Information Theory*, 59(1):472–481, 2013.

R. Arora, A. Cotter, and N. Srebro. Stochastic optimization of PCA with capped MSG. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

P. Assouad. Deux remarques sur l'estimation. *Comptes rendus des séances de l'Académie des sciences. Série 1, Mathématique*, 296(23):1021–1024, 1983.

M.F. Balcan and V. Feldman. Statistical active learning algorithms for noise tolerance and differential privacy. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

M.F. Balcan, A. Blum, S. Fine, and Y. Mansour. Distributed learning, communication complexity and privacy. In *Conference on Learning Theory (COLT)*, 2012.

Z. Bar-Yossef, T.S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Science*, 68: 702–732, 2004.

S. Ben-David and E. Dichterman. Learning with restricted focus of attention. *Journal of Computer and System Sciences*, 56:277–298, 1998.

A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 253–262. ACM, 1994.

A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003.

M. Braverman and A. Rao. Information equals amortized communication. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011.

M. Braverman, A. Garg, T. Ma, H.L. Nguyen, and D.P. Woodruff. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. *arXiv preprint arXiv:1506.07216*, 2015.

A. Chakrabarti, Y. Shi, A. Wirth, and A. Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2001.

G. Cormode and S Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

T.M. Cover. Hypothesis testing with finite statistics. *The Annals of Mathematical Statistics*, 40 (3):828–835, 1969.

J. Duchi, M. Jordan, and M. Wainwright. Local privacy and statistical minimax rates. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2013.

V. Feldman. A complete characterization of statistical query learning with applications to evolvability. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2009.

V. Feldman, E. Grigorescu, L. Reyzin, S. Vempala, and Y. Xiao. Statistical algorithms and a lower bound for detecting planted cliques. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*. ACM, 2013.

A. Ganor, G. Kol, and R. Raz. Exponential separation of information and communication. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2014.

A. Garg, T. Ma, and H. Nguyen. On communication cost of distributed statistical estimation and dimensionality. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

A. Gilbert and P. Indyk. Sparse recovery using sparse matrices. *Proceedings of the IEEE*, 98(6): 937–947, 2010.

A. Gupta, M. Hardt, A. Roth, and J. Ullman. Privately releasing conjunctions and the statistical query barrier. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, 2011.

M.E. Hellman and T.M. Cover. Learning with finite memory. *The Annals of Mathematical Statistics*, 41(3):765–782, 1970.

S.P. Kasiviswanathan, H.K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6): 983–1006, 1998.

J. Kivinen and M.K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, New York, NY, USA, 1997.

J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10:777–801, 2009.

L. Le Cam. *Asymptotic methods in statistical theory.* Springer-Verlag, New York, 1986.

T. Lee and A. Shraibman. Lower bounds in communication complexity. *Foundations and Trends in Theoretical Computer Science*, 3(4):263398, 2009.

I. Mitliagkas, C. Caramanis, and P. Jain. Memory limited, streaming PCA. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

F. Niu, B. Recht, C. Re, and S. Wright. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

R. Raz. Fast learning requires good memory: A time-space lower bound for parity learning. *Electronic Colloquium on Computational Complexity (ECCC)*, 2016.

O. Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4), 2008.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

O. Shamir. Fundamental limits of online and distributed algorithms for statistical learning and estimation. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

J. Steinhardt and J. Duchi. Minimax rates for memory-bounded sparse linear regression. In *Conference on Learning Theory (COLT)*, 2015.

J. Steinhardt, S. Wager, and P. Liang. The statistics of streaming sparse regression. *arXiv preprint arXiv:1412.4182*, 2014.

B. Szörényi. Characterizing statistical query learning: simplified notions and proofs. In *Algorithmic Learning Theory*, pages 186–200. Springer, 2009.

A.B. Tsybakov. *Introduction to Nonparametric Estimation.* Springer, 2009.

L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010.

Y. Zhang, J. Duchi, M. Jordan, and M. Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

A Proofs for Section 2

A.1 Proof of Proposition 3

The high level proof approach is to argue that we can simulate all of the communications just by using statistical queries. We will proceed via induction, and argue that the total variation distance between the distribution of communications in the communication model, and the distribution of *simulated* communications based on statistical queries, is small—bounded by α—and hence if the communication algorithm succeeds with probability $1-\delta$, the statistical query algorithm will be successful with probability at least $1-\delta-\alpha$. The following protocol describes how each step of this simulation proceeds.

Algorithm 1. Simulating Communication Protocol via Statistical Queries
Given the description of a communication protocol in which each of m players receives an example x drawn from p, the following algorithm describes how to simulate the protocol using statistical queries.

Consider an intermediate step of the communication protocol in which the i-th player is supposed to communicate the next bit. Assume that the i-th player has already communicated $j-1$ bits, $c_1, \ldots, c_{j-1}$ in earlier steps of the protocol. Let $f_{i,j}$ denote the (possibly randomized) function that maps x to $\{0,1\}$, representing the function that player i uses to compute the j-th bit c_j to communicate. Note that $f_{i,j}$ might have been chosen dependent on the entire transcript of communications up to this point, and let $f_{i,1}, \ldots, f_{i,j-1}$ denote the analogous functions that were used to compute the first $j-1$ bits that player i communicated (which were each dependent on the transcripts of communication up until the corresponding bits were communicated).

For $k \leq j-1$, let $E_k = \bigwedge_{h=1}^{k}[f_{i,h}(x) = c_h]$ and $p_k = \Pr_x[E_k]$, and assume that we have estimates of $p_1, \ldots, p_{j-1}$, which we denote by $q_1, \ldots, q_{j-1}$, satisfying $|p_k - q_k| \leq 2\tau$.

1. *We ask two statistical queries of tolerance τ:*

$$
\begin{aligned}
{}_0q_j &:= SQ\Big(\mathbb{I}[E_{j-1} \wedge f_{i,j}(x) = 0], \tau\Big), \\
{}_1q_j &:= SQ\Big(\mathbb{I}[E_{j-1} \wedge f_{i,j}(x) = 1], \tau\Big),
\end{aligned}
$$

   where $\mathbb{I}$ is the indicator function.

2. *Define $t = \frac{{}_0q_j + (q_{j-1} - {}_1q_j)}{2}$ and $s = \max(0, \min(t, q_{j-1}))$.*
3. *Set $c_j = 0$ with probability s/q_{j-1}, and $c_j = 1$ otherwise.*
4. *If $c_j = 0$, then set $q_j = s$, otherwise set $q_j = q_{j-1} - s$.*

We note that in the above reduction, in Step 1 we use two statistical queries—one to estimate ${}_0q_j \approx \Pr[E_{j-1} \wedge f_{i,j}(x) = 0]$ and one to estimate ${}_1q_j \approx \Pr[E_{j-1} \wedge f_{i,j}(x) = 1]$. We could have gotten away with a single query, leveraging the fact that ${}_0q_j + {}_1q_j \approx q_{j-1}$, though the errors in the approximation would increase as more bits are communicated, yielding that $|p_k - q_k| \leq k\tau$, rather than the invariant that $|p_k - q_k| \leq 2\tau$. Hence we could have reduced the total number of statistical queries by a factor of 2, at the expense of needing to decrease the tolerance by a factor of b—the total number of bits communicated per player.

The following lemma justifies the calculations in Step 2 of the above algorithm, showing that the above process preserves the invariant that $|p_k - q_k| \leq 2\tau$.

Lemma 18. *Assume ${}_0p_j, {}_1p_j, p_{j-1} \in [0,1]$ satisfy ${}_1p_j + {}_0p_j = p_{j-1}$, and let ${}_0q_j, {}_1q_j$, and q_{j-1} satisfy $|p_{j-1} -$*

$q_{j-1}| \leq 2\tau$, $|{}_0p_j -_0 q_j| \leq \tau$, and $|{}_1p_j - {}_1q_j| \leq \tau$. Then if we define

$$t = \frac{{}_0q_j + (q_{j-1} - {}_1q_j)}{2} \text{ and } s = \max\left(0, \min\left(t, q_{j-1}\right)\right),$$

the following two inequalities hold:

$$|{}_0p_j - s| \leq 2\tau \text{ and } |{}_1p_j - (q_{j-1} - s)| \leq 2\tau.$$

Proof. First note that since $0 \leq {}_0p_j, {}_1p_j \leq p_{j-1}$, and $|p_{j-1} - q_{j-1}| \leq 2\tau$, it suffices to show that $|{}_0p_j - t| \leq 2\tau$ and $|{}_1p_j - (q_{j-1} - t)| \leq 2\tau$, as the restriction of t to lie in the range $[0, q_{j-1}]$ can never cause these equations to go from being true to being false. For the first inequality, note that $q_{j-1} - {}_1q_j \in [{}_1p_j - 3\tau, {}_0p_j - 3\tau]$, and hence ${}_0q_j + (q_{j-1} - {}_1q_j) \in [2{}_0p_j - 4\tau, 2{}_0p_j + 4\tau]$, from which the first inequality follows. For the second inequality, first note that $({}_0q_j - {}_1q_j) - ({}_0p_j - {}_1p_j)| \leq 2\tau$, and hence $({}_0q_j - {}_1q_j) - (p_{j-1} - 2{}_1p_j)| \leq 2\tau$. Plugging this in, we have the following:

$$\begin{aligned}
\left| {}_1p_j - \left(q_{j-1} - \frac{{}_0q_j + (q_{j-1} - {}_1q_j)}{2} \right) \right| &= \left| {}_1p_j + \frac{-q_{j-1} + ({}_0q_j - {}_1q_j)}{2} \right| \\
&\leq \left| {}_1p_j + \frac{-q_{j-1} + p_{j-1} - 2{}_1p_j}{2} \right| + \tau, \\
&\leq \left| \frac{p_{j-1} - q_{j-1}}{2} \right| + \tau \\
&\leq 2\tau.
\end{aligned}$$

□

We are now equipped to prove the validity of the simulation algorithm, establishing Proposition 3.

Proof of Proposition 3. Let c_t denote the bit of communication communicated in round t, and let i_t and j_t denote the corresponding party and index of the bit. For shorthand, define the tuple $z_t = (i_t, j_t, c_t)$, and let $m' = bm$ denote the total number of rounds of communication. Then, we can bound the total variational distance between the true distribution $p(z_{1:m'})$ and the simulated distribution $q(z_{1:m'})$ as

$$\begin{aligned}
\|p - q\|_{TV} &= \frac{1}{2} \sum_{z_{1:m'}} |p(z_{1:m'}) - q(z_{1:m'})| \\
&\leq \frac{1}{2} \sum_{t=1}^{m'} \mathbb{E}_{z_{1:t-1}} \left[|p(i_t, j_t \mid z_{1:t-1}) - q(i_t, j_t \mid z_{1:t-1})| \right] \\
&\qquad\qquad + \mathbb{E}_{z_{1:t-1}, i_t, j_t} \left[|p(c_t \mid i_t, j_t, c_{1:t-1}) - q(c_t \mid i_t, j_t, c_{1:t-1})| \right] \\
&= \frac{1}{2} \sum_{t=1}^{m'} \mathbb{E}_{z_{1:t-1}, i_t, j_t} \left[|p(c_t \mid c_{1:t-1}, i_t, j_t) - q(c_t \mid c_{1:t-1}, i_t, j_t)| \right],
\end{aligned}$$

where the final equality is because i_t is selected using the same protocol for both p and q, and j_t is a deterministic function of $(z_{1:t-1}, i_t)$.

Consequently, for a given party i, the contribution from the simulation of bit c_j to the total variation distance is bounded by $\frac{1}{2}\mathbb{E}\left[\left|\frac{p_j}{p_{j-1}} - \frac{q_j}{q_{j-1}}\right|\right]$. We now analyze this quantity. First observe that

$$\mathbb{E}\left[\frac{1}{q_j} \mid q_{j-1}\right] = u \frac{1}{u q_{j-1}} + (1 - u) \frac{1}{(1 - u) q_{j-1}} = \frac{2}{q_{j-1}},$$

where $u = s/q_{j-1}$ is the probability of the coin used to decide c_j as specified in Step 3 of the algorithm. Hence $\frac{1}{2q_1}, \frac{1}{2^2 q_2}, \ldots, \frac{1}{2^j q_j}$ is a martingale, from which it follows that $\mathbb{E}[\frac{1}{q_j}] = 2^j$, where the expectation is taken over the randomness of the entire simulation.

From Lemma 18, $|q_k - p_k| \leq 2\tau$, hence we have the following inequality:

$$\begin{aligned}\left|\frac{p_j}{p_{j-1}} - \frac{q_j}{q_{j-1}}\right| &= \frac{|p_j q_{j-1} - q_j p_{j-1}|}{p_{j-1}q_{j-1}} \\ &\leq \frac{p_j|q_{j-1} - p_{j-1}| + p_{j-1}|p_j - q_j|}{p_{j-1}q_{j-1}} \\ &\leq \frac{2\tau(p_j + p_{j-1})}{p_{j-1}q_{j-1}} \leq \frac{4\tau p_{j-1}}{p_{j-1}q_{j-1}} \leq \frac{4\tau}{q_{j-1}}.\end{aligned}$$

Consequently, $\frac{1}{2}\mathbb{E}\left[\left|\frac{p_j}{p_{j-1}} - \frac{q_j}{q_{j-1}}\right|\right] \leq 2^j\tau$, and so the overall contribution of the ith party to the total variational distance is at most $\sum_{j=1}^{b} 2^j\tau < 2^{b+1}\tau$. Summing over all parties yields a bound of $\|p-q\|_{TV} < m2^{b+1}\tau$. Hence, as long as $\tau \leq \frac{\delta}{m2^{b+1}}$, the total variational distance is bounded by δ, which means that the probability of failure can increase by at most δ, and so is bounded by 2δ, as desired. □

A.2 Proof of Proposition 6

We will simply simulate whatever statistical queries would have been made, using only a small number of samples and 1 bit of communication per sample.

More precisely, we will show that we can with probability $1 - \frac{\delta}{m}$ simulate a single statistical query of tolerance τ, using $\frac{2\log(2\delta/m)}{\tau^2}$ samples and 1 bit of communication per sample. By the union bound, we can then simulate m statistical queries with probability $1 - \delta$.

To simulate a single statistical query, suppose we have k samples $x_1, \ldots, x_k$. For each sample i, evaluate $\psi_i = \psi(x_i)$, and round it to $+1$ with probability $\frac{1+\psi_i}{2}$ and to -1 with probability $\frac{1-\psi_i}{2}$. If we let z_i be the result of this rounding process, then clearly $\mathbb{E}[z_i] = \mathbb{E}_{x\sim\mathcal{D}}[\psi(x)]$. Moreover, if we let $\overline{z} = \frac{1}{k}(z_1 + \cdots + z_k)$, then by the Chernoff bound, we have

$$\mathbb{P}[|\overline{z} - \mathbb{E}[\psi(x)]| \geq \tau] \leq 2\exp(-k\tau^2/2). \tag{13}$$

For $k \geq \frac{2\log(2m/\delta)}{\tau^2}$, this probability is less than $\frac{\delta}{m}$, as desired. □

A.3 Proof of Corollary 5

In this section we prove Corollary 5, which shows that even if we have a multiparty communication protocol in which each party is given a set of k examples drawn from PARITY(n), then if each party communicates at most b bits, $2^{(n-k-2b)/3-7}/(bn)$ parties are needed to $(1/5, 1/32)$-learn the parity.

While the techniques of Blum et al. [2003] already yield an exponential statistical query lower bound for parity with k examples, they incur constants in the exponent that we wish to avoid here. Instead, we will make rely on the machinery developed in this paper.

First, suppose that one can $(1/3, 1/32)$-learn k-example parity with $2^{(n-k-2b)/3-7}/(bn)$ parties. By Proposition 3, one can then $(1/3, 1/16)$-learn k-example parity with $2^{(n-k-2b)/3-6}/n$ queries of tolerance $bn/2^{(n-k+b)/3-1}$. Note also that if the parity is noisy with noise level $\frac{1}{2}$, we can accomodate this by decreasing the tolerance by a factor of 2. Then, by Proposition 6, one can $(1/3, 1/8)$-learn k-example noisy parity in the one-way communication model with $(2^{(n-k-2b)/3-3}/n)\log(2^{(n-k-2b)/3}/n) \cdot (2^{2(n-k+b)/3-2}/b^2n^2) \leq 2^{n-k-5}/(3n^2b^2)$ parties and 1 bit of communication. Finally, once we have learned a parity function with probability $\frac{7}{8}$, we can distinguish a noisy parity function from random noise with overall probability greater than $\frac{3}{4}$ by simply drawing $\mathcal{O}(1)$ additional examples and checking that sufficiently many of them match the learned parity (for instance, 34 additional examples suffice, if we check that at least 22 of the examples match).

On the other hand, by Theorem 11 and the remarks following Corollary 12, we know that no one-way 1-bit communication algorithm can learn noisy parity with fewer than 2^{n-k-4} k-ary examples and probability greater than $\frac{3}{4}$. Consequently, we must have $2^{n-k-5}/(3n^2b^2) + 34 \geq 2^{n-k-4}$, which is impossible unless $n - k \leq 10$, in which case the result is vacuously true.

B Proofs for Section 3

B.1 Proof of Theorem 7

The proof of Theorem 7 relies on a reduction of statistical queries to a canonical form. First, we replace statistical queries with *statistical threshold queries*, which return a binary yes-no answer about whether a statistical query lies above or below a given threshold. We also apply a normalization procedure to ensure that at least one of the two answers will narrow down the space of concepts by a factor of 1/2 or better. Then, we simply record the first point at which we receive an "important" answer that narrows the space by at least this factor: by construction, all of the previous answers are then uniquely determined, so that this is enough to recover the full sequence of queries and responses up to that point. Iterating this process $\mathcal{O}(\log|\mathcal{F}|)$ times leaves us with a unique concept.

To make the canonicalization process more precise, recall that a statistical query oracle SQ takes a statistic $\psi : \mathcal{X} \to [-1, 1]$ and a tolerance τ, and outputs a value $\mu = \mathrm{SQ}(\psi, \tau)$ satisfying $|\mu - \mathbb{E}_{x\sim\mathcal{D}}[\psi(x)]| < \tau$. We define a related *statistical threshold query* oracle, which takes a triple (ψ, t, τ) and outputs a response $r = \mathrm{STQ}(\psi, t, \tau) \in \{0, 1\}$, such that $r = 1$ if $\mathbb{E}_{x\sim\mathcal{D}}[\psi(x)] \geq t + \tau$, $r = 0$ if $\mathbb{E}_{x\sim\mathcal{D}}[\psi(x)] \leq t - \tau$, and r may be arbitrary otherwise. Then, for any statistical threshold query $q = (\psi, t, \tau)$, define $\mathcal{F}^1(q)$ and $\mathcal{F}^0(q)$ to be the subsets of $\mathcal{F}$ consistent with a response of 1 and 0 to the query, respectively:

$$\mathcal{F}^1(q) \overset{\text{def}}{=} \{f \in \mathcal{F} : \mathbb{E}_{x\sim\mathcal{D}}[\psi(x)] > t - \tau\}, \tag{14}$$

$$\mathcal{F}^0(q) \overset{\text{def}}{=} \{f \in \mathcal{F} : \mathbb{E}_{x\sim\mathcal{D}}[\psi(x)] < t + \tau\}. \tag{15}$$

Note that $\mathcal{F}^1(q) \cup \mathcal{F}^0(q) = \mathcal{F}$. We say that a statistical threshold query $q = (\psi, t, \tau)$ is *valid* if

$$\min\left\{\left|\mathcal{F}^1(q)\right|, \left|\mathcal{F}^0(q)\right|\right\} \leq \frac{1}{2}|\mathcal{F}|. \tag{16}$$

As shown in the following result, any statistical query can be replaced with a small number of valid statistical threshold queries.

Lemma 19. *Any statistical query with tolerance $\tau \leq 2$ can be implemented with $\lceil \log\left(\frac{2}{\tau}\right)\rceil$ statistical threshold queries with tolerance $\tau/2$. Moreover, any statistical threshold query with tolerance $\tau/2$ can be simulated with a valid statistical threshold query with tolerance $\tau/4$.*

Now, for a query q, call a response r *light* if $|\mathcal{F}^r(q)| \leq \frac{1}{2}|\mathcal{F}|$, and *heavy* otherwise. If we get a light response to q, then we can simply remember this response and recursively solve the problem for the class $\mathcal{F}^r(q)$, which is at least 50% smaller than before. If we have only received heavy responses so far, then the sequence of responses and queries is uniquely determined (since there is at most one heavy response to each query and the algorithm is deterministic) and we can remember this with 1 bit of memory. This yields:

Lemma 20. *If a problem is $(\varepsilon, 0)$-learnable with $m = m_0\lceil\log\left(\frac{2}{\tau}\right)\rceil$ statistical threshold queries with tolerance $\tau/2$, then it can be $(\varepsilon, 0)$-learned with $\mathcal{O}(m\log|\mathcal{F}|)$ statistical threshold queries with tolerance $\tau/4$ and $\mathcal{O}(\log|\mathcal{F}|\log(m))$ bits of memory.*

The $k = 1$ case of Theorem 7 follows by using the fact that all of the statistical threshold queries in Lemma 19 can be obtained from a single statistical query of tolerance $\tau/4$, and that $\mathcal{O}m_0 \log|\mathcal{F}|$ such statistical queries can be simulated with probability $1 - \delta$ using $\mathcal{O}\left(m_0 \log|\mathcal{F}|\log(m_0 \log|\mathcal{F}|/\delta)/\tau^2\right)$ samples.

To extend Theorem 7 to general k, we simply instead simulate k statistical queries simultaneously on the same $\log(m/\delta)/\tau^2$ samples. Since the queries are adaptive, this requires "knowing" the future. However, we can use the following idea: simulate the queries that we would have made assuming that all k of the next responses are heavy. If they are indeed all heavy, then our simulation was successful. If they are not, then we can take the first light response, remember it, and recurse on a concept class of half the size (which can happen at most $\log|\mathcal{F}|$ times). This reduces the total number of statistical queries needed from $\mathcal{O}(m_0 \log|\mathcal{F}|)$ to $\mathcal{O}\left(\lceil\frac{m_0}{k}\rceil\log|\mathcal{F}|\right)$.

Beyond the memory requirements of Lemma 20, we also need storage to simulate the statistical queries themselves; each statistical query requires $\mathcal{O}(\log(1/\tau))$ bits (to store the running averages to accuracy τ), and so introduces $k\log(1/\tau)$ additional bits of storage.

Overall, then, we end up with $\mathcal{O}(\log|\mathcal{F}|\log(m_0\log(1/\tau)))$ storage from Lemma 20, and $k\log(1/\tau)$ storage from simulating the queries, and require $\mathcal{O}\left(\lceil m_0/k\rceil\log|\mathcal{F}|\log(m_0\log|\mathcal{F}|/\delta)/\tau^2\right)$ samples, as claimed.

B.2 Proof of Lemma 19

We first show that any statistical query with tolerance $\tau \leq 2$ can be implemented with $\left\lceil \log\left(\frac{2}{\tau}\right)\right\rceil$ statistical threshold queries with tolerance $\frac{\tau}{2}$. To do so, we use the following binary search algorithm to implement $\mathrm{SQ}(\psi, \tau)$:

```
$L \leftarrow -1 - \tau/2$
$R \leftarrow 1 + \tau/2$
while $R - L > 2\tau$ do
    $M \leftarrow \frac{L+R}{2}$
    $b \leftarrow \mathrm{STQ}(\psi, M, \tau/2)$
    if $b = 1$ then
        $L \leftarrow M - \tau/2$
    else
        $R \leftarrow M + \tau/2$
    end if
end while
return $\frac{L+R}{2}$
```

Consider the value of $R - L$; it is initially $2 + \tau$, and on each loop iteration updates as $(R-L)_{\text{new}} = \frac{1}{2}(R-L)_{\text{old}} + \frac{\tau}{2}$. After m loop iterations (and hence m calls to $\mathrm{STQ}(\psi, \cdot, \tau/2)$) we thus have $R - L = (2+\tau) \cdot 2^{-m} + \tau(1 - 2^{-m}) = \tau + 2^{1-m}$. For $m = \left\lceil \log_2\left(\frac{2}{\tau}\right)\right\rceil$, we then have $R - L \leq 2\tau$, at which point the loop exits. But we also then know that $\mathbb{E}[\psi(x)] \in (L, R)$, whence $\frac{R+L}{2}$ is a valid output of $\mathrm{SQ}(\psi, \tau)$.

Next, we show that any statistical threshold query with tolerance τ can be simulated with a valid statistical threshold query with tolerance at least $\tau/2$. Consider any statistical threshold query $q = (\psi, t, \tau)$. We claim that at least one of $q_0 = (\psi, t - \tau/2, \tau/2)$ and $q_1 = (\psi, t + \tau/2, \tau/2)$ is valid. Indeed, $\mathcal{F}_0(q_0)$ and $\mathcal{F}_1(q_1)$ are disjoint, so at least one must have size at most $\frac{1}{2}|\mathcal{F}|$. Furthermore, both q_0 and q_1 yield strictly more information than q, and so we can simulate q from either q_0 or q_1. $\square$

B.3 Proof of Lemma 20

We formalize our procedure with the following pseudocode.

```
procedure BoundedMemoryLearn($\mathcal{F}$, $\mathcal{D}$):
if $|\mathcal{F}| = 1$ then
    return the unique element of $\mathcal{F}$
end if
for $i = 1$ to $m$ do
    $q \leftarrow$ queryIfAllHeavy($i$, $\mathcal{F}$, $\mathcal{D}$) {gets query $i$ assuming all previous responses are heavy}
    if not valid($q$, $\mathcal{F}$, $\mathcal{D}$) then
        $q \leftarrow$ makeValid($q$, $\mathcal{F}$, $\mathcal{D}$)
    end if
    $r \leftarrow \mathrm{STQ}(q)$
    if light($r$, $\mathcal{F}$, $\mathcal{D}$) then
        $firstLightIndex \leftarrow i$
        $lightResponse \leftarrow r$
        return BoundedMemoryLearn($\mathcal{F}^r(q)$, $\mathcal{D}$) {$q$ is the first query to get a light response}
    end if
end for
return answerIfAllHeavy($\mathcal{F}$, $\mathcal{D}$)
```

Note that when we recurse to subsets of $\mathcal{F}$, we can use the same statistical query algorithm as before (since if it learns $\mathcal{F}$ it will certainly learn any subset of $\mathcal{F}$), though the return values of `valid` and `light` may change. By construction, the above procedure can only recurse $\mathcal{O}(\log|\mathcal{F}|)$ times, and only requires m queries and $\mathcal{O}(\log(m))$ bits at each level of recursion (to keep track of i, *firstLightIndex*, and *lightResponse*).[4]

[4] Note that, even though `answerIfAllHeavy`($\mathcal{F}$, $\mathcal{D}$) need not lie in $\mathcal{F}$, it is a fixed quantity depending only on $\mathcal{F}$ and $\mathcal{D}$, and so requires no additional bits to represent.

This establishes the result. □

B.4 Proof of Theorem 8

To prove Theorem 8, it suffices to provide an efficient SQ learning algorithm for linear regression and then apply Theorem 7. To this end, we note the following standard result on performing gradient descent with small errors on the gradients:

Lemma 21. *Let f be a convex function. Suppose that for any point w with $\|w\|_1 \leq R$ ($R \geq 1$), one can obtain an approximate gradient z satisfying $\|z\|_\infty \leq B$ and $\|z - \nabla f(w)\|_\infty \leq \tau B$. Then, after m approximate gradient computations, exponentiated gradient obtains an estimate $\hat{w}$ satisfying*

$$f\left(\hat{w}\right) - \min_{\|w\|_1 \leq R} f(w) \leq \mathcal{O}\left(\sqrt{\frac{RB^2 \log(n)}{m}} + \tau RB\right). \tag{17}$$

The proof of Theorem 8 now follows by taking $f(w) = \mathbb{E}_{x\sim\mathcal{D}}[(w \cdot x - y)^2]$; the gradient $\nabla f(w)$ is equal to $\mathbb{E}_{x\sim\mathcal{D}}[2x(w \cdot x - y)]$, which has coordinates bounded by $B = \mathcal{O}\left(k\right)$. Then, $\nabla f(w)/B$ is a mean of random variables over $[-1,\, 1]^n$ and so can be estimated with n statistical queries; in particular, queries with tolerance τ lead to an accuracy of τB for the estimate z of $\nabla f(w)$. Also note that we can take $R = \mathcal{O}(k)$. Thus, by Lemma 21, we obtain a vector $\hat{w}$ satisfying $f(\hat{w}) - f(w^*) \leq \varepsilon$ in

$$M = \mathcal{O}\left(n\varepsilon^{-2} \log\left(n\right) k^3\right) \text{ statistical queries of tolerance } \tau = \mathcal{O}\left(\varepsilon \,/\, k^2\right), \tag{18}$$

whether or not w^* is k-sparse. Now, if in addition w^* is known to be k-sparse, the answer w^* can be represented (to accuracy ε) with $\log|\mathcal{F}| = \mathcal{O}\left(k \log\left(n/\varepsilon\right)\right)$ bits of memory. We immediately conclude from Theorem 7 that the k-sparse linear regression problem can be $(\varepsilon,\, \delta)$-solved with

$$b = \mathcal{O}\left(k \log^2\left(\frac{n}{\varepsilon}\right) + \log\left(\frac{k^2}{\varepsilon}\right)\right) \text{ bits of memory and } M = \widetilde{\mathcal{O}}\left(\frac{nk^8}{\varepsilon^4} \log\left(\frac{1}{\delta}\right)\right) \text{ samples,} \tag{19}$$

as claimed.

B.5 Proof of Lemma 21

We make use of the following version of exponentiated gradient, which restricts the domain of optimization to the l^1-ball of radius R; the vector γ_j measures the error in the approximation z_j:

$$z_j = \nabla f(w_j) + \gamma_j, \tag{20}$$

$$w_{j+1} = \arg\min_w \left\{\eta^{-1} \sum_{i=1}^n w^{(i)} \log\left(w^{(i)}\right) + \left\langle w,\, \sum_{j'=1}^j z_{j'}\right\rangle : \|w\|_1 \leq R,\, w \geq 0\right\}. \tag{21}$$

Note that this algorithm restricts to the positive orthant $w \geq 0$; however, we can remove this assumption (while making our constants worse by a factor of 2) by splitting each coordinate of w into a positive and negative part [Kivinen and Warmuth, 1997].

The approximate gradient z_j can be interpreted as the exact gradient of the modified function $\tilde{f}(w) = f(w) + \gamma_j \cdot w$. Thus, by standard online convex optimization results [e.g., Shalev-Shwartz, 2011], for any $\eta > 0$ and $n \geq 3$ we have

$$\begin{aligned}
&\sum_{j=1}^m \left(f(w_j) - f(w^*) + \gamma_j \cdot (w_j - w^*)\right) \\
&\quad \leq \eta^{-1}\left(R\log(n/R) + \sum_{i=1}^n w^{*,(i)} \log\left(w^{*,(i)}\right)\right) + \eta \sum_{j=1}^m \|z_j\|_\infty^2 \\
&\quad \leq \eta^{-1}\left(R\log(n) - R\log(R) + \|w^*\|_1 \log\|w^*\|_1\right) + \eta m B^2 \\
&\quad \leq \eta^{-1} R \log\left(n\right) + \eta m B^2,
\end{aligned}$$

where the last inequality uses the fact that $x\log(x) \le y\log(y)$ whenever $x \le y$ and $y \ge 1$. Optimizing our choice of η, we then get

$$\sum_{j=1}^{m} \left(f(w_j) - f(w^*) + \gamma_j \cdot (w_j - w^*)\right) \le 2\sqrt{mR\log(n)}B.$$

Moreover, since $\|w_j - w^*\|_1 \le 2R$, we know that $\gamma_j \cdot (w_j - w^*) \ge -2\tau RB$, and so

$$\sum_{j=1}^{m} \left(f(w_j) - f(w^*)\right) \le 2\sqrt{mR\log(n)}B + 2\tau RBm.$$

Finally, the desired conclusion follows by convexity of $f(\cdot)$.

B.6 Proof of Theorem 9

Our proof relies on the min-count sketch construction [Cormode and Muthukrishnan, 2005]. For our purposes, the main implication of the construction is as follows: there is a distribution of matrices $A \in \{0, 1\}^{\omega d \times n}$ such that, for any fixed vector $u \in \mathbb{R}^n$, we can recover $\hat{u}$ from Au such that (see, e.g., Section 2 of Gilbert and Indyk [2010])

$$\|\hat{u} - u\|_\infty \le \frac{2}{\omega}\|u\|_1, \text{ with probability at least } 1 - n\,2^{-d}.$$

Moreover, if we only get to observe Au up to tolerance τ in each coordinate, resulting in a recovered vector $\hat{u}_\tau$, then the result correspondingly weakens to

$$\|\hat{u}_\tau - u\|_\infty \le \frac{2}{\omega}\|u\|_1 + \tau, \text{ with probability at least } 1 - n\,2^{-d}.$$

In our situation, we want to estimate $z^* = \nabla f(w)$ as defined in Lemma 21 with as few statistical queries as possible. To do this, we proceed as follows (letting $u = z^*$):

- Draw a matrix A from the count-sketch distribution with parameters d and ω,
- Estimate Az^* to tolerance $\tau\|Az^*\|_\infty$ using ωd statistical queries of tolerance τ, and finally
- Generate a recovered vector z using the count-min sketch algorithm.

Note that $\|Az^*\|_\infty \le \|z^*\|_1$ (since all entries of A are in $\{0,1\}$) and hence the error on Az^* is at most $\tau\|z^*\|_1 \le \tau\|x\|_1|y - w\cdot x| \le \mathcal{O}(\tau rk)$. We of course also have $\|u\|_1 = \|z^*\|_1 \le \mathcal{O}(rk)$. Toegether, these imply that $\|z - z^*\|_\infty \le \mathcal{O}\left(\left(\frac{2}{\omega} + \tau\right) rk\right)$. Thus, setting

$$\omega = \left\lceil \frac{1}{\tau} \right\rceil \text{ and } d = \left\lceil \log(n) + \log\left(\delta^{-1}\right) + \log(m) \right\rceil, \text{ we conclude that } \|z_m - z_m^*\|_\infty \le \mathcal{O}(\tau rk), \tag{22}$$

with probability at least $1 - \delta/m$; by the union bound, (22) holds with probability at least $1 - \delta$ for all m samples simultaneously. The upshot is that we can now get the gradients z from $\omega d = \mathcal{O}(\log(n/\delta)/\tau)$ statistical queries instead of n queries. In summary, we can then obtain error ε using

$$\mathcal{O}\left(k^3\log(n/\delta)/(\tau\varepsilon^2)\right) \text{ queries of tolerance } \tau = \mathcal{O}\left(\varepsilon/rk^2\right). \tag{23}$$

To apply Theorem 7, we then need to derandomize, which we can do by standard amplification techniques, at the cost of increasing the number of queries by a further factor of $k\log(n/\varepsilon)$. Finally, applying Theorem 7, the number of samples needed is (up to polylogarithmic factors in k, n, and ε) the number of statistical queries divided by the square of the tolerance, times $\log(1/\delta)$. This yields a final number of samples equal to

$$\widetilde{\mathcal{O}}\left(r^3k^{10}\log^2(1/\delta)/\varepsilon^5\right), \tag{24}$$

as claimed.

C Direct Communication Bounds with Assouad's Method

C.1 Proof of Lemma 13

We use a standard result on the total variation distance $\|p-q\|_{TV} = \frac{1}{2}\int |p(x)-q(x)|dx$, which says that it is impossible to distinguish between distributions p and q with probability greater than $\frac{1+\|p-q\|_{TV}}{2}$ [Le Cam, 1986]. Applying this to the distributions p_0 and $\frac{1}{n}\sum_{i=1}^{n} p_i$, we obtain:

$$\begin{aligned}
&\left\| \frac{1}{n}\sum_{i=1}^{n} p_i - p_0 \right\|_{TV} \\
&\leq \frac{1}{n}\sum_{i=1}^{n} \|p_i - p_0\|_{TV} \\
&\leq \frac{1}{n}\sum_{i=1}^{n} \sqrt{\mathrm{KL}\left(p_0(Z^{(1:m)}) \parallel p_i(Z^{(1:m)})\right)/2} && \text{(Pinsker's inequality)} \\
&\leq \sqrt{\frac{1}{n}\sum_{i=1}^{n} \mathrm{KL}\left(p_0(Z^{(1:m)}) \parallel p_i(Z^{(1:m)})\right)/2} && \text{(Jensen's inequality)} \\
&= \sqrt{\frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{E}_{\hat{z}=z^{(1:j-1)}\sim p_0}\left[\mathrm{KL}\left(p_0(Z^{(j)} \mid \hat{z}) \parallel p_i(Z^{(j)} \mid \hat{z})\right)\right]} && \text{(chain rule)} \\
&\leq \sqrt{\frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{E}_{\hat{z}=z^{(1:j-1)}\sim p_0}\left[D_{\chi^2}\left(p_0(Z^{(j)} \mid \hat{z}) \middle\| p_i(Z^{(j)} \mid \hat{z})\right)\right]} \\
&= \sqrt{D/2},
\end{aligned}$$

where the final inequality is Lemma 2.7 of Tsybakov [2009]. The overall probability of success is thus at most $\frac{1}{2} + \sqrt{D/8}$, as claimed. □

C.2 Proof of Lemma 14

Recall that Z is a potentially random function of $X = X^{(j)}$ and $\hat{Z}$; writing $\pi(z \mid x, \hat{z})$ for its conditional distribution, we have that

$$p_0(z \mid \hat{z}) = \sum_{x\in\mathcal{X}} \pi(z \mid x, \hat{z})\, p_0(x), \quad p_i(z \mid \hat{z}) = \sum_{x\in\mathcal{X}} \pi(z \mid x, \hat{z})\, p_i(x).$$

We then have:

$$\begin{aligned}
\sum_{i=1}^{n}(p_i(z \mid \hat{z}) - p_0(z \mid \hat{z}))^2 &= \sum_{i=1}^{n}\left(\sum_{x\in\mathcal{X}} \pi(z \mid x, \hat{z})(p_i(x) - p_0(x))\right)^2 \\
&= \sum_{i=1}^{n}\sum_{x,x'\in\mathcal{X}} \pi(z \mid x, \hat{z})(p_i(x) - p_0(x))(p_i(x') - p_0(x'))\pi(z \mid x', \hat{z}) \\
&= \sum_{x,x'\in\mathcal{X}} \pi(z \mid x, \hat{z}) \left[\sum_{i=1}^{n}(p_i(x) - p_0(x))(p_i(x') - p_0(x'))\right] \pi(z \mid x', \hat{z}) \\
&= \sum_{x,x'\in\mathcal{X}} \pi(z \mid x, \hat{z})\sqrt{p_0(x)} \left[\sum_{i=1}^{n} \frac{p_i(x) - p_0(x)}{\sqrt{p_0(x)}} \frac{p_i(x') - p_0(x')}{\sqrt{p_0(x')}}\right] \sqrt{p_0(x')}\pi(z \mid x', \hat{z}) \\
&\leq \lambda_{\max}(\hat{M}) \sum_{x\in\mathcal{X}} p_0(x)\pi(z \mid x, \hat{z})^2 \\
&\leq \lambda_{\max}(\hat{M}) \sum_{x\in\mathcal{X}} p_0(x)\pi(z \mid x, \hat{z}) \\
&= \lambda_{\max}(\hat{M})\, p_0(z \mid \hat{z}), \qquad (25)
\end{aligned}$$

where $\hat{M}_{x,x'} \overset{\text{def}}{=} \sum_{i=1}^{n} \frac{p_i(x)-p_0(x)}{\sqrt{p_0(x)}} \frac{p_i(x')-p_0(x')}{\sqrt{p_0(x')}}$, and the last step follows since $\pi(z|x,\hat{z}) \leq 1$. Finally, we note that $\hat{M}$ is the companion matrix to $M_{ij} = \sum_{x\in\mathcal{X}} \frac{(p_i(x)-p_0(x))(p_j(x)-p_0(x))}{p_0(x)}$, and hence $\lambda_{\max}(\hat{M}) = \lambda_{\max}(M)$, from which the result follows.

C.3 Proof of Lemma 15

We first verify that

$$\begin{aligned}
M_{ij}^{(k)} + 1 &= \sum_{x_1,\dots,x_k\in\mathcal{X}} \frac{p_i(x_1)p_j(x_1)\cdots p_i(x_k)p_j(x_k)}{p_0(x_1)\cdots p_0(x_k)} \qquad (26) \\
&= \prod_{l=1}^{k}\sum_{x_l\in\mathcal{X}} \frac{p_i(x_l)p_j(x_l)}{p_0(x_l)} \qquad (27) \\
&= \prod_{l=1}^{k}(M_{ij} + 1) \qquad (28) \\
&= (M_{ij} + 1)^k. \qquad (29)
\end{aligned}$$

Let $A \odot B$ be the element-wise product of matrices A and B. We use the following two facts about positive semidefinite matrices: (1) $\lambda_{\max}(A + B) \leq \lambda_{\max}(A) + \lambda_{\max}(B)$, and (2) $A \odot B$ is positive semidefinite and $\lambda_{\max}(A \odot B) \leq \lambda_{\max}(A) \cdot \lambda_{\max}(B)$. Then, since M is positive semidefinite, we have

$$\begin{aligned}
\lambda_{\max}(M^{(k)}) &= \lambda_{\max}\left(\sum_{i=1}^{k}\binom{k}{i}M^{\odot i}\right) \qquad (30) \\
&\leq \sum_{i=1}^{k}\binom{k}{i}\lambda_{\max}(M^{\odot i}) \qquad (31) \\
&\leq \sum_{i=1}^{k}\binom{k}{i}\lambda_{\max}(M)^i \qquad (32) \\
&= (\lambda_{\max}(M) + 1)^k - 1, \qquad (33)
\end{aligned}$$

as was to be shown.

C.4 Computations with χ^2- and KL-divergence.

In this section we prove:

Lemma 22. *Let p_i denote a parity with noise $\varepsilon = \frac{1}{2}$, and let p_0 denote uniformly random noise. Suppose that we store a single example. Then, the KL divergence $\mathrm{KL}\left(p_0 \parallel p_i\right)$ is $\frac{1}{2}\log(4/3)$, and the χ^2-divergence $D_{\chi^2}\left(p_i \| p_0\right)$ is $\frac{1}{3}$.*

Suppose instead that we store $4n$ examples with probability $\frac{1}{4n}$. Then the expected KL divergence $\mathrm{KL}\left(p_0 \parallel p_i\right)$ is still $\frac{1}{2}\log(4/3)$, but the expected χ^2-divergence $D_{\chi^2}\left(p_0 \| p_i\right)$ is $\frac{1}{4n}\left(\left(\frac{4}{3}\right)^{4n} - 1\right)$.

Proof. In the first case, the KL divergence is

$$\sum_x p_0(x)\log(p_0(x)/p_i(x)). \tag{34}$$

If x is a positive example under p_i, then $p_i(x) = \frac{(1-\varepsilon)}{2^{n-1}} + \frac{\varepsilon}{2^n}$; otherwise it is just $\frac{\varepsilon}{2^n}$. On the other hand, $p_0(x)$ is equal to $\frac{1}{2^n}$ always. Therefore, argument to the $\log(\cdot)$ term is $\frac{1}{\varepsilon}$ half of the time, and $\frac{1}{2-\varepsilon}$ the other half of the time. The KL divergence is therefore $\frac{1}{2}\log\left(\frac{1}{\varepsilon(2-\varepsilon)}\right)$, which is $\frac{1}{2}\log(4/3)$ when $\varepsilon = \frac{1}{2}$.

The χ^2-divergence is

$$\sum_x p_0(x)\frac{p_0(x)}{p_i(x)} - 1. \tag{35}$$

Again, p_0/p_i is $\frac{1}{\varepsilon}$ half the time and $\frac{1}{2-\varepsilon}$ half the time. On average, it is then $\frac{1}{2}\left(2 + \frac{2}{3}\right) = \frac{4}{3}$; subtracting 1 yields the claimed result of $\frac{1}{3}$.

In the second case, the KL divergence does not change in expectation; this is because with probability $\frac{4n-1}{4n}$, the KL divergence is zero, and with probability $\frac{1}{4n}$, we get the KL divergence between $4n$ independent copies of the same distributions as before. Since KL divergence is additive across the independent copies, the two factors of $4n$ cancel.

On the other hand, as we saw in Lemma 15, the χ^2-divergence of independent copies behaves as $D_{\chi^2}\left(p^{\otimes k} \middle\| q^{\otimes k}\right) = \left(D_{\chi^2}\left(p \| q\right) + 1\right)^k - 1$. Threfore, the χ^2-divergence is $\frac{1}{4n}\left(\left(\frac{4}{3}\right)^{4n} - 1\right)$, as claimed. □

C.5 Proof of Proposition 17

Suppose that we have an algorithm that takes m steps. We will perform step $i \in \{1, \ldots, m\}$ on the $(i-1)k + r$-th sample, where $r \sim \mathrm{Uniform}(\{1, \ldots, k\})$. For all other samples we transmit a blank message.

Let us now compute the entropy of each message; each message is either blank (with probability $1 - 1/k$) or else is a message whose entropy is $b = \mathrm{poly}(n)$ (due to communicating the entire sample). Letting $h_2(p)$ be the entropy of a coin flip with probability p, we can then bound the total entropy of each message by

$$\begin{aligned}
\frac{b}{k} + h_2(1/k) &= \frac{b}{k} + \frac{1}{k}\log_2(k) + \frac{k-1}{k}\log_2\left(\frac{k}{k-1}\right) && (36)\\
&= \frac{b}{k} + \frac{\log_2(k)}{k} + \frac{1}{k}\log_2\left(\left(1 + \frac{1}{k-1}\right)^{k-1}\right) && (37)\\
&\leq \frac{b + \log_2(k) + \log_2(e)}{k}. && (38)
\end{aligned}$$

Letting $k = \omega(b)$, we can make the per-message entropy arbitrarily small, and in particular less than 1, as was to be shown. □

C.6 Proof of Proposition 16

We will assume that we have an algorithm that outputs a $\hat{p}$ satisfying $\rho(p, \hat{p}) \leq \varepsilon$, and construct a new algorithm that achieves error at most 2ε, and which uses very little memory.

We split our memory into three chunks; at a high level, the first chunk will be used to run an instance of the original algorithm, the second chunk will be used to store the answer, and the third chunk will store a small amount of auxiliary data.

Our procedure first samples a random number $j \in \{1, \ldots, k\}$. Then, it runs the original algorithm on the samples with index $i \in \{(j-1)m+1, \ldots, jm\}$ (using the third chunk to track j and i). Afterwards, it writes the recovered answer to the second chunk, and zeroes out the first chunk. Once the index i reaches km, it returns the answer in the second chunk (which is guaranteed to have been written to exactly once).

First, to see that this algorithm works, we need to make sure that the answer can be represented with $\log_2 |\mathcal{F}|$ bits. This is straightforward if the answer $\hat{p}$ lies in $\mathcal{F}$, but this need not be the case. On the other hand, we know that $\rho(p, \hat{p}) \leq \varepsilon$ for our distance metric ρ. Therefore, take any $p' \in \mathcal{F}$ satisfying $\rho(p', \hat{p}) \leq \varepsilon$, and use this as the answer[5]; this p' will satisfy $\rho(p, p') \leq 2\varepsilon$ by the triangle inequality, which is all that we require.

Now, let us measure the entropy of z_i under this procedure. As before, the first chunk is zeroes with probability $1 - \frac{1}{k}$, and is otherwise a random variable with entropy at most $b = \mathrm{poly}(n)$. We can then conclude that the entropy of this chunk is at most $\frac{\log_2(k \cdot e)}{k} + \frac{b}{k}$ (see (38) above). Furthermore, the second chunk takes on at most $|\mathcal{F}|$ values and the third chunk at most $k \cdot km$ values (for the random draw j together with the counter i), and so they together add at most $\log_2(k^2 m|\mathcal{F}|)$ bits to the entropy. Setting $k = b$, the total entropy is $\frac{\log_2(b \cdot e)}{b} + 1 + \log_2(b^2 m|\mathcal{F}|)$, where b and m are both $\mathrm{poly}(n)$. The claimed result therefore follows.

[5] Finding such a p' may in general require superpolynomial computation.