Time-Space Tradeoffs for Learning Finite Functions from Random Evaluations, with Applications to Polynomials

Paul Beame[*] BEAME@CS.WASHINGTON.EDU
Shayan Oveis Gharan[†] SHAYAN@CS.WASHINGTON.EDU
Xin Yang[*] YX1992@CS.WASHINGTON.EDU
University of Washington

Editors: Sebastien Bubeck, Vianney Perchet and Philippe Rigollet

Abstract

We develop an extension of recent analytic methods for obtaining time-space tradeoff lower bounds for problems of learning from uniformly random labelled examples. With our methods we can obtain bounds for learning concept classes of finite functions from random evaluations even when the sample space of random inputs can be significantly smaller than the concept class of functions and the function values can be from an arbitrary finite set.

At the core of our results, we reduce the time-space complexity of learning from random evaluations to the question of how much the corresponding evaluation matrix amplifies the 2-norms of "almost uniform" probability distributions. To analyze the latter, we formulate it as a semidefinite program, and we analyze its dual. In order to handle function values from arbitrary finite sets, we apply this norm amplification analysis to complex matrices.

As applications that follow from our new techniques, we show that any algorithm that learns n-variate polynomial functions of degree at most d over $\mathbb{F}_2$ with success at least $2^{-O(n)}$ from evaluations on randomly chosen inputs either requires space $\Omega(nm/d)$ or $2^{\Omega(n/d)}$ time where $m = (n/d)^{\Theta(d)}$ is the dimension of the space of such polynomials. These bounds are asymptotically optimal for polynomials of arbitrary constant degree since they match the tradeoffs achieved by natural learning algorithms for the problems. We extend these results to learning polynomials of degree at most d over any odd prime field $\mathbb{F}_p$ where we show that $\Omega((mn/d)\log p)$ space or time $p^{\Omega(n/d)}$ is required.

To derive our bounds for learning polynomials over finite fields, we show that an analysis of the dual of the corresponding semidefinite program follows from an understanding of the distribution of the bias of all degree d polynomials with respect to uniformly random inputs.

Keywords: Memory-bounded learning. Lower bounds.

1. Introduction

In supervised learning from labelled examples, the question of the sample complexity required to obtain good generalization has been of considerable interest and research. However, another important parameter is how much information from these samples needs to be kept in memory in

[*] Research supported in part by NSF grant CCF-1524246
[†] Research supported in part by NSF grant CCF-1552097 and ONR-YI grant N00014-17-1-2429

order to learn successfully. There has been a line of work improving the memory efficiency of learning algorithms, and the question of the limits of such improvement has begun to be tackled relatively recently. Shamir (2014) and Steinhardt et al. (2016) both obtained constraints on the space required for certain learning problems and in the latter paper, the authors asked whether one could obtain strong tradeoffs for learning from random samples that yields a superlinear threshold for the space required for efficient learning. Raz (2016) showed that even given exact information, if the space of a learning algorithm is bounded by a sufficiently small quadratic function of the number of input bits, then the problem of online of learning parity functions given exact answers on random samples requires an exponential number of samples even to learn parity functions approximately.

More precisely, we consider problems of online learning from uniform random samples, in which an unknown concept x is chosen uniformly from a set X of (multivalued) concepts and a learner is given a stream of samples $(a^{(1)}, b^{(1)}, (a^{(2)}, b^{(2)}), \cdots$ where each $a^{(t)}$ is chosen uniformly at random from A and $b^{(t)} = L(a^{(t)}, x)$ for labelling function L which maps each pair (a, x) to the outcome (or label) of the value of concept $x \in X$ when given $a \in A$. The learner's goal is either that of identification "find x" or prediction "predict $L(a, x)$ for randomly chosen a with significant advantage over random guessing." In the case of learning parities, $X = A = \{0,1\}^n$ and $L(a, x) = a \cdot x \pmod 2$. With high probability $n + 1$ uniformly random samples suffice to span $\{0,1\}^n$ and one can learn parities using Gaussian elimination with $(n+1)^2$ space. Alternatively, an algorithm with only $O(n)$ space can wait for a specific basis of vectors a to appear (for example the standard basis) and store the resulting values; however, this takes $\Omega(2^n)$ time. Raz (2016) showed that either $\Omega(n^2)$ space or $2^{\Omega(n)}$ time is essential: even if the space is bounded by $n^2/25$, $2^{\Omega(n)}$ queries are required to learn x correctly with any probability that is $2^{-o(n)}$. In follow-on work, Kol et al. (2017) showed that the same lower bound applies even if the input x is sparse.

We can view x as a linear function over $\mathbb{F}_2$, and, from this perspective, parity learning identifies a linear function from evaluations over uniformly random inputs. A natural generalization asks if a similar lower bound exists when we learn higher degree polynomials with bounded space. As a motivating example, consider homogeneous quadratic functions over $\mathbb{F}_2$. Let $m = \binom{n+1}{2}$ and $X = \{0,1\}^m$, which we identify with the space of homogeneous quadratic polynomials in $\mathbb{F}_2[z_1, \ldots, z_n]$ or, equivalently, the space of upper triangular Boolean matrices. This learning algorithm has $A = \{0,1\}^n$ and $X = \{0,1\}^m$, and the learning function $L(a, x) = x(a) = \sum_{i \leqslant j} x_{ij} a_i a_j \bmod 2$, or equivalently $L(a, x) = a^T x a$ when x is viewed as an upper triangular matrix.

Given $a \in \{0,1\}^n$ and $x \in \{0,1\}^m$, we can view evaluating $x(a)$ as computing $aa^T \cdot x \bmod 2$ where we interpret aa^T as an element of $\{0,1\}^m$. For $O(m)$ randomly chosen $a \in \{0,1\}^n$, the vectors aa^T almost surely span $\{0,1\}^m$ and hence we can store m samples of the form (a, b) and apply Gaussian elimination to determine x. This time, we only need $n + 1$ bits to store each sample for a total space bound of $O(nm)$. An alternative algorithm using $O(m)$ space and time $2^{O(n)}$ would be to look for a specific basis such as the basis consisting of the upper triangular parts of $\{e_i e_i^T \mid 1 \leqslant i \leqslant n\} \cup \{(e_i + e_j)(e_i + e_j)^T \mid 1 \leqslant i < j \leqslant n\}$. Previous lower bounds for learning do not apply to this problem[1] because $|A| \ll |X|$. Our results imply that this tradeoff between $\Omega(nm)$ space or $2^{\Omega(n)}$ time is inherently required to learn x with probability $2^{-o(n)}$ or predict its output with at least $2^{-o(n)}$ advantage.

[1] Note that in Kol et al. (2017) the lower bound applies in a dual case when the unknown x is sparse, and hence $|X| \ll |A|$.

The techniques in Raz (2016) and Kol et al. (2017) were based on fairly ad-hoc simulations of the original space-bounded learning algorithm by a restricted form of linear branching program for which one can measure progress at learning x using the dimension of the consistent subspace. More recent papers, by Moshkovitz and Moshkovitz (2017) using graph mixing properties and by Raz (2017) using an analytic approach, considered more general tests and used a measure of progress based on 2-norms. While the method of Moshkovitz and Moshkovitz (2017) was not strong enough to reproduce the bound in Raz (2016) for the case of parity learning, the methods of Raz (2017) and the later improvement (Moshkovitz and Moshkovitz, 2018) of Moshkovitz and Moshkovitz (2017) reproduced the parity learning bound and more.

In particular, Raz (2017) defined a ± 1 matrix M that is indexed by $A \times X$. It is natural to see $M(a,x)$ as $(-1)^{L(a,x)}$ for a labelling function L that has labels in $\{0,1\}$. The lower bound is governed by the (expectation) matrix norm of M, which is a function of the largest singular value of M, and the progress is analyzed by bounding the impact of applying the matrix to probability distributions with small expectation 2-norm. This method works if $|A| \geqslant |X|$ - i.e., the sample space of inputs is at least as large as the concept class - but it fails completely if $|A| \ll |X|$, which is precisely the situation for learning quadratic functions. Indeed, none of the prior approaches works in this case.

In our work we extend the analytic approach to capture *general* discrete problems of learning from uniform random samples in which (1) the sample space of inputs can be much smaller than the concept class and (2) members of the concept class can have values from an arbitrary finite set, which we identify with $\{0,1,\ldots,r\}$ for convenience. Our extensions come from two different directions.

We define a property of matrices M that allows us to refine the notion of the largest singular value and extend the method of Raz (2017) to the cases that $|A| \ll |X|$. This property, which we call the *norm amplification curve* of the matrix on the positive orthant, analyzes more precisely how $\|M \cdot p\|_2$ grows as a function of $\|p\|_2$ for probability vectors p on X. The key reason that this is not simply governed by the singular values is that the interior of the positive orthant can contain at most one singular vector. We give a simple condition on the 2-norm amplification curve of M that is sufficient to ensure that there is a time-space tradeoff showing that any learning algorithm for M with success probability at least $2^{-\varepsilon n}$ for some $\varepsilon > 0$ either requires space $\Omega(mn)$ or time $2^{\Omega(n)}$.

For any fixed learning problem given by a matrix M, the natural way to express the amplification curve at any particular value of $\|p\|_2$ yields an optimization problem given by a quadratic program with constraints on $\|p\|_2^2$, $\|p\|_1$ and $p \geqslant 0$, and with objective function $\|Mp\|_2^2 = \langle M^T M, pp^T \rangle$ that seems difficult to solve. Instead, we relax the quadratic program to a semi-definite program where we replace pp^T by a positive semidefinite matrix U with the analogous constraints. We can then obtain an upper bound on the amplification curve by moving to the SDP dual and evaluating the dual objective at a particular Laplacian determined by the properties of $M^T M$.

In order to handle concepts that are more than binary-valued[2], we move to matrices whose entries are complex r-th roots of unity. Indeed, a single matrix M does not suffice for $r > 3$ and we instead work with a family of complex matrices $M^{(1)}, \ldots, M^{(r-1)}$, each associated with a different

[2] The formalization of Moshkovitz and Moshkovitz (2017, 2018) does include the case of multivalued outcomes, though they do not apply it to any examples and their mixing condition does not hold in the case of small input sample spaces

root of unity. We use the natural generalization of the norm amplification curve to complex matrices and also generalize the semi-definite relaxation method to bound these curves using $(M^{(j)})^*M^{(j)}$ instead of M^TM. We then show how the overall analytic approach can be carried through with a modest number of changes from the binary-valued case.

Our lower bound shows that if the 2-norm amplification curve for M has (or, in the case of r-valued labels, matrices $M^{(1)},\ldots,M^{(r-1)}$ have) the required property, then to achieve learning success probability for M of at least $|A|^{-\varepsilon}$ for some $\varepsilon > 0$, either space $\Omega(\log|A| \cdot \log_r|X|)$ or time $|A|^{\Omega(1)}$ is required. This matches the natural upper bounds asymptotically over a wide range of learning problems.

As applications, we focus on problems of learning polynomials of varying degrees over finite fields. For matrices M associated with polynomials over $\mathbb{F}_2$, the property of the matrices M^TM required to bound the amplication curves for M correspond to properties of the weight distribution of Reed-Muller codes over $\mathbb{F}_2$. For quadratic polynomials, this weight distribution is known exactly. In the case of higher degree polynomials, bounds on the weight distribution of such codes, or more precisely on the bounds on the bias of random degree d polynomials over $\mathbb{F}_2$ of Ben-Eliezer et al. (2012) are sufficient to let us show that learning polynomials of degree at most d over $\mathbb{F}_2^n$ from random inputs with probability $2^{-\Omega(n/d)}$ either requires space $\Omega(nm/d)$ or time $2^{\Omega(n/d)}$.

We also analyze learning problems for the case of prime fields $\mathbb{F}_p$ for p odd using our multivalued techniques involving complex matrices. For $\mathbb{F}_p$, even the cases of linear and affine polynomials are new. We relate the norm amplification curves of the associated matrices to bounds on the bias of random degree d polynomials over $\mathbb{F}_p$. We also give a precise analysis of the bias of the set of quadratic polynomials over $\mathbb{F}_p^n$ to derive tight time-space tradeoff lower bounds for learning them. For larger degrees we apply bounds on the bias that we recently proved in a companion paper (Beame et al., 2018b).

Related work: Independently and contemporaneously with our preliminary version (Beame et al., 2017), Garg et al. (2017) proved closely related results to ours for the case of binary labels. The fundamental techniques are similarly grounded in the approach of Raz (2017). At the very high-level, they prove very similar structural properties of the matrix M, namely, they show that it is an "L_2 two-source extractor" which can be seen to be equivalent to bounding our norm amplification curve for learning matrices. More precisely, their "almost orthogonality property" essentially corresponds to upper bounding $W_\kappa(M^*M)$ for some threshold κ (see Definition 9 and Lemma 10). However, since we use duality explicitly, our proof seems more amenable to extensions, particularly, when we have more structure in the learning matrix M. Subsequently, Garg et al. (2018) were also able to allow multivalued labels by extending the extractor approach to permit correlations between the sample inputs and the concept. Our full paper appears as Beame et al. (2018a).

2. Branching programs for learning

In order to be able to solve the learning problem given concept class X, sample space of inputs A and labelling function L on $A \times X$ exactly we require that the learning function L have the property that for all $x \neq x' \in X$ there exists an $a \in A$ such that $L(a,x) \neq L(a,x')$. Note that the set $\{0,1,\ldots,r-1\}$ of labels allows us to model any learning situation in which r different labels are possible.

Following Raz (2016), the time and space of a learner are modelled simultaneously by expressing the learner's computation as a layered branching program: a finite rooted directed acyclic multigraph with every non-sink node having outdegree $r|A|$, with one outedge for each (a,b) with $a \in A$ and $b \in \{0,1,\ldots,r-1\}$ that leads to a node in the next layer. Each sink node v is labelled by some $x' \in X$ which is the learner's guess of the value of the concept x. (In the case of prediction we allow the sink label to be an arbitrary function from A to $\{0,1,\ldots,r-1\}$ denoting the best prediction of the algorithm for each $a \in A$.)

The space S used by the learning branching program is the $\log_2$ of the maximum number of nodes in any layer and the time T is the length of the longest path from the root to a sink.

The samples given to the learner $(a^{(1)},b^{(1)}),(a^{(2)},b^{(2)}),\ldots$ based on uniformly randomly chosen $a^{(1)},a^{(2)},\ldots \in A$ and a concept $x \in X$ determines a (randomly chosen) *computation* path in the branching program. When we consider computation paths we include the concept x in their description. The (expected) success probability of the learner is the probability for a uniformly random concept $x \in X$ that a random computation path given concept x reaches a sink node v with label $x' = x$ (or with sufficiently good predictions on randomly chosen $a \in A$).

Progress towards identification Following Moshkovitz and Moshkovitz (2017) and Raz (2017) we measure progress towards identifying $x \in X$ using the "expectation 2-norm" over the uniform distribution: For any set S, and $f : S \to \mathbb{R}$, define $\|f\|_2 = \left(\mathbb{E}_{s\in_R S} f^2(s)\right)^{1/2} = (\sum_{s\in S} f^2(s)/|S|)^{1/2}$. Define Δ_X to be the space of probability distributions on X. Consider the two extremes for the expectation 2-norm of elements of Δ_X: If $\mathbb{P}$ is the uniform distribution on X, then $\|\mathbb{P}\|_2 = |X|^{-1}$. This distribution represents the learner's knowledge of the concept x at the start of the branching program. On the other hand if $\mathbb{P}$ is point distribution on any x', then $\|\mathbb{P}\|_2 = |X|^{-1/2}$.

For each node v in the branching program, there is an induced probability distribution on X, $\mathbb{P}'_{x|v}$ which represents the distribution on X conditioned on the fact that the computation path passes through v. It represents the learner's knowledge of x at the time that the computation path has reached v. Intuitively, the learner has made significant progress towards identifying the concept x if $\|\mathbb{P}'_{x|v}\|_2$ is much larger than $|X|^{-1}$, say $\|\mathbb{P}'_{x|v}\|_2 \geqslant |X|^{\delta/2} \cdot |X|^{-1} = |X|^{-(1-\delta/2)}$.

The general idea will be to argue that for any fixed node v in the branching program that is at a layer t that is $|A|^{o(1)}$, the probability over a randomly chosen computation path that v is the first node on the path for which the learner has made significant progress is $|X|^{-\Omega(\log_r |A|)}$. Since by assumption of correctness the learner makes significant progress with at least $|X|^{-\varepsilon}$ probability, there must be at least $|X|^{\Omega(\log_r |A|)}$ such nodes and hence the space must be $\Omega(\log |X| \log_r |A|)$.

Given that we want to consider the first vertex on a computation path at which significant progress has been made, it is natural to truncate a computation path at v if significant progress has been already been made at v (and then one should not count any path through v towards the progress at some subsequent node w). Following Raz (2017), for technical reasons we will also truncate the computation path in other circumstances: if the concept x has too high probability at v, or if the next edge is labelled by a pair (a,b) for which the value on input a of random concepts whose computation path reaches v is significantly biased away from the uniform distribution on $\{0,1\ldots,r-1\}$.

Like Raz (2017), we use an analytic approach to understanding the progress and the bias. In Raz (2017), only binary feedback is possible and progress is analyzed in terms of the matrix properties of a learning matrix M given by $M(a,x) = (-1)^{L(a,x)}$, which is viewed as the learning problem specification. This form is particularly convenient since it allows one to represent the predictability of outcomes under a distribution $\mathbb{P}$ on X in terms of a matrix vector product. That is, $(M \cdot \mathbb{P})(a) = \mathbf{E}_{x \sim \mathbb{P}}[(-1)^{L(a,x)}]$ is the expected bias of a concept distributed according to $\mathbb{P}$ on input a.

It would be natural to try to extend this analytic approach for $r > 2$ by replacing $(-1)^{L(a,x)}$ by $\omega^{L(a,x)}$ where $\omega = e^{2\pi i/r}$ is a primitive r-th root of unity. However, for $r > 3$, simply having $\mathbf{E}_{x \in_R X}[\omega^{f(x)}]$ small does not imply that f is close to uniformly distributed on $\{0, 1, \ldots, r-1\}$. Nonetheless, by setting $g_k = \mathbf{Pr}_{x \in_R X}[f(x) = k]$ we can apply the following proposition, which is a standard method using exponential sums, to show that bounding $|\mathbf{E}_{x \in_R X}[\omega^{j \cdot f(x)}]|$ for all $j \in \{1, \ldots, r-1\}$ is sufficient to show that f is close to uniformly distributed.

Proposition 1 *Suppose that $\sum_{k=0}^{r-1} g_k = 1$ and define $g(z) = \sum_{k=0}^{r-1} g_k z^k$. If $|g(\omega^j)| < \varepsilon$ for all $j \in \{1, \ldots, r-1\}$ then for all $k \in \{0, 1, \ldots, r-1\}$, $|g_k - \frac{1}{r}| \leqslant \varepsilon$.*

Therefore, instead of the single ± matrix M given by $M(a,x) = (-1)^{L(a,x)}$, we will analyze the learning problem given by L using $r-1$ different[3] complex matrices $M^{(j)} \in \mathbb{C}^{A \times X}$ for $j \in \{1, \ldots, r-1\}$ given by $M^{(j)}(a,x) = \omega^{j \cdot L(a,x)}$. We now define the probability distributions and truncation process for computation paths inductively as follows:

Definition 2 *We define probability distributions $\mathbb{P}_{x|v} \in \Delta_X$ and the (δ, α, γ)-truncation of the computation paths inductively as follows:*

- *If v is the root, then $\mathbb{P}_{x|v}$ is the uniform distribution on X.*
- *(Significant Progress) If $\|\mathbb{P}_{x|v}\|_2 \geqslant |X|^{-(1-\delta/2)}$ then truncate all computation paths at v. We call vertex v* significant *in this case.*
- *(High Probability) Truncate the computation paths at v for all concepts x' for which $\mathbb{P}_{x|v}(x') \geqslant |X|^{-\alpha}$. Let $\mathrm{High}(v)$ be the set of such concepts.*
- *(High Bias) Truncate any computation path at v if it follows an outedge e of v with label (a,b) for which $|(M^{(j)} \cdot \mathbb{P}_{x|v})(a)| \geqslant |A|^{-\gamma}$ for some $j \in \{1, \ldots, r-1\}$. That is, we truncate the paths at v if the label outcome for the next sample for $a \in A$ is too predictable given the knowledge that the path was not truncated previously and arrived at v.*
- *If v is not the root then define $\mathbb{P}_{x|v}$ to be the conditional probability distribution on x over all computation paths that have not previously been truncated and arrive at v.*

For an edge $e = (v,w)$ of the branching program, we also define a probability distribution $\mathbb{P}_{x|e} \in \Delta_X$, which is the conditional probability distribution on X induced by the truncated computation paths that pass through edge e.

With this definition, it is no longer immediate from the assumption of correctness that the truncated path reaches a significant node with at least $|A|^{-\varepsilon}$ probability. However, we will see that a single assumption about the matrices $M^{(j)}$ will be sufficient to prove both that this holds and that the probability is $|X|^{-\log_r |A|}$ that the path reaches any specific node v at which significant progress has been made.

[3] In Proposition 1 one can observe that $|g(\omega^j)| = |g(\omega^{r-j})|$ so $\lceil (r-1)/2 \rceil$ matrices suffice, but we find it convenient to argue using all $r-1$ matrices; however, this does imply that a single matrix suffices when $r = 3$.

3. Norm amplification by matrices on the positive orthant

By definition, for $\mathbb{P} \in \Delta_X$, and $M \in \mathbb{C}^{A\times X}$, $\|M \cdot \mathbb{P}\|_2^2 = \mathbf{E}_{a\in_R A}[|(M \cdot \mathbb{P})(a)|^2]$. Observe that for $\mathbb{P} = \mathbb{P}_{x|v}$ and $M = M^{(j)}$ for $j \in \{1, \dots, r-1\}$, the values $|(M^{(j)} \cdot \mathbb{P}_{x|v})(a)|$ are the quantities that we test to determine whether an edge labelled a is a high bias edge that causes the truncation of the computation path. Therefore $\|M^{(j)} \cdot \mathbb{P}_{x|v}\|_2^2$ is the expected square of this bias value for uniformly random inputs at v.

If we have not learned the concept x, we would not expect to be able to predict its value on a random input; moreover, since any path that would follow a high bias input is truncated, it is essential to argue that $\|M^{(j)} \cdot \mathbb{P}_{x|v}\|_2$ remains small at any node v where there has not been significant progress.

In Raz (2017) there is a single ± 1 matrix M and $\|M \cdot \mathbb{P}_{x|v}\|_2$ is bounded using the matrix norm $\|M\|_2$ given by $\|M\|_2 = \sup_{\substack{f:X\to\mathbb{R}\\ f\neq 0}} \|M \cdot f\|_2/\|f\|_2$, where the numerator is an expectation 2-norm over A and the denominator is an expectation 2-norm over X. Thus $\|M\|_2 = \sqrt{|X|/|A|} \cdot \sigma_{\max}(M)$, where $\sigma_{\max}(M)$ is the largest singular value of M and $\sqrt{|X|/|A|}$ is a normalization factor.

In the case of the matrix M associated with parity learning, $|A| = |X| = 2^n$ and all the singular values are equal to $\sqrt{|X|}$ so $\|M\|_2 = \sqrt{|X|} = 2^{n/2}$. With this bound, if v is not a node of significant progress then $\|\mathbb{P}_{x|v}\|_2 \leqslant 2^{-(1-\delta/2)n}$ and hence $\|M \cdot \mathbb{P}_{x|v}\|_2 \leqslant 2^{-(1-\delta)n/2}$ which is $1/|A|^{(1-\delta)/2}$ and hence small.

However, even in the case of learning quadratic functions over $\mathbb{F}_2$, the largest singular value of the matrix M is still $\sqrt{|X|}$ (the uniform distribution on X is a singular vector) and so $\|M\|_2 = |X|/\sqrt{|A|}$. But in that case, when $\|\mathbb{P}_{x|v}\|_2$ is $|X|^{-(1-\delta/2)}$ we conclude that $\|M\|_2 \cdot \|\mathbb{P}_{x|v}\|_2$ is at most $|X|^{\delta/2}/\sqrt{|A|}$ which is much larger than 1 and hence a useless bound on $\|M \cdot \mathbb{P}_{x|v}\|_2$.

Indeed, the same kind of problem occurs in using the method of Raz (2017) for any learning problem for which $|A|$ is $|X|^{o(1)}$: If v is a child of the root of the branching program at which the more likely outcome b of a single randomly chosen input $a \in A$ is remembered, then $\|\mathbb{P}_{x|v}\|_2 \leqslant \sqrt{2}/|X|$. However, in this case $|(M \cdot \mathbb{P}_{x|v})(a)| = 1$ and so $\|(M \cdot \mathbb{P}_{x|v})\|_2 \geqslant |A|^{-1/2}$. It follows that $\|M\|_2 \geqslant |X|/(2|A|)^{1/2}$ and when $|A|$ is $|X|^{o(1)}$ the derived upper bound on $\|M \cdot \mathbb{P}_{x|v'}\|_2$ at nodes v' where $\|\mathbb{P}_{x|v'}\|_2 \geqslant 1/|X|^{1-\delta/2}$ will be larger than 1 and therefore useless.

We need a more precise way to bound $\|M \cdot \mathbb{P}\|_2$ as a function of $\|\mathbb{P}\|_2$ than using the single number $\|M\|_2$. To do this we will need to use the fact that $\mathbb{P} \in \Delta_X$ – it has a fixed ℓ_1 norm and (more importantly) it is non-negative and therefore lies in the positive orthant.

Definition 3 *For $M \in \mathbb{C}^{A\times X}$ the 2*-norm amplification curve *of M, $\tau_M : [0,1] \to \mathbb{R}$ is given by*

$$\tau_M(\delta) = \sup_{\substack{\mathbb{P}\in\Delta_X\\ \|\mathbb{P}\|_2 \leqslant 1/|X|^{1-\delta/2}}} \log_{|A|}(\|M \cdot \mathbb{P}\|_2).$$

In other words, whenever $\|\mathbb{P}\|_2$ is at most $|X|^{-(1-\delta/2)}$, $\|M \cdot \mathbb{P}\|_2$ is at most $|A|^{\tau_M(\delta)}$. To prove our lower bounds we will bound the norm amplification curves $\tau_{M^{(j)}}$ for all $j \in \{1, \dots, r-1\}$.

4. Theorems

Our general lower bound for learning problems over arbitrary finite label sets is given by following theorem.

Theorem 4 *There are constants $c_1, c_2, c_3 > 0$ such that the follow holds. Let $L : A \times X \to \{0, 1, \ldots, r-1\}$ be a labelling function and for $j = 1, \cdots, r-1$ define the matix $M^{(j)} \in \mathbb{C}^{A\times X}$ by $M^{(j)}(a, x) = \omega^{j\cdot L(a,x)}$ where $\omega = e^{2\pi i/r}$ and assume*[4] *that $|A| \leqslant |X|$. Suppose that for $0 < \delta' < 1$ we have $\tau_{M^{(j)}}(\delta') \leqslant -\gamma' < 0$ for all $j \in \{1, \cdots, r-1\}$. Then, for $\varepsilon \geqslant c_1 \min(\delta', \gamma') > 0$, $\beta \geqslant c_2 \min(\delta', \gamma') > 0$, and $\eta \geqslant c_3 \delta' \gamma' > 0$, any algorithm that solves the learning problem for L with success probability at least $|A|^{-\varepsilon}$ or advantage $\geqslant |A|^{-\varepsilon/2}$ either requires space at least $\eta \log_2 |A| \log_r |X|$ or time at least $|A|^{\beta}$.*

Applications to learning polynomials There are many potential applications of the above theorem but for this paper we focus learning polynomials from their evaluations over finite fields of various sizes. The bounds are derived using the semidefinite programming approach given in Section 6 together with analyses for polynomials given in the full paper (Beame et al., 2018a).

Learning polynomials over $\mathbb{F}_2$ We first consider the case of polynomials over $\mathbb{F}_2$ which yield a binary labelling set. In this case $\omega = -1$ and there is only one matrix M whose entries are $M(a, x) = (-1)^{L(a,x)}$ as in Raz (2017).

The case of linear functions over $\mathbb{F}_2$ is just the parity learning problem. For learning higher degree polynomials over $\mathbb{F}_2$ we obtain the following bounds on the norm amplification curves of their associated matrices:

Theorem 5 *The following properties hold:*
(a) For all $\delta \in [0, 1]$, the matrix M for learning quadratic functions over $\mathbb{F}_2^n$ satisfies

$$\tau_M(\delta) \leqslant \tfrac{-(1-\delta)}{8} + \tfrac{5+\delta}{8n}.$$

(b) For any $\zeta > 0$, there are constants δ, γ with $0 < \delta < 1/2$ and $\gamma > 0$ such that for $d \leqslant (1-\zeta)n$ the matrix M for learning functions of degree $\leqslant d$ over $\mathbb{F}_2^n$ satisfies $\tau_M(\delta) \leqslant -\gamma/d$.

The case for quadratic polynomials over $\mathbb{F}_2$ follows from properties of the weight distribution of Reed-Muller codes $RM(n, 2)$ shown by Sloane and Berlekamp (1970) and McEliece (1967). The case for higher degree polynomials over $\mathbb{F}_2$ follows from tail bounds on the bias of $\mathbb{F}_2$ polynomials given by Ben-Eliezer et al. (2012).

Using these bounds together with Theorem 4 yields the following:

Theorem 6 *There are constants $\varepsilon, \zeta > 0$ such that the following hold:*
(a) Let $m = \binom{n+1}{2}$ for positive integer n. Any algorithm for learning quadratic functions over $\mathbb{F}_2^n$ that succeeds with probability at least $2^{-\varepsilon n}$ requires space $\Omega(nm)$ or time $2^{\Omega(n)}$.
(b) Let $n > 0$ and $d > 0$ be integers such that $d \leqslant (1-\zeta)\cdot n$ and let $m = \sum_{i=0}^{d} \binom{n}{i}$. Any algorithm for learning polynomial functions of degree at most d over $\mathbb{F}_2^n$ that succeeds with probability at least $2^{-\varepsilon n/d}$ requires space $\Omega(nm/d)$ or time $2^{\Omega(n/d)}$.

[4] We could write the statement of the theorem to apply to all A and X by replacing each occurrence of $|A|$ in the lower bounds with $\min(|A|, |X|)$. When $|A| \geqslant |X|$ and $r = 2$, we can use $\|M\|_2$ to bound $\tau_M(\delta')$ which yields the bound given in Raz (2017)

These bounds are tight for constant d since they match the resources used by the natural learning algorithms described in the introduction up to constant factors in the space bound and in the exponent of the time bound.

Learning polynomials over $\mathbb{F}_p$ for odd prime p. The following theorem bounds the norm amplification curves for polynomials of various degrees over odd prime fields.

Theorem 7 *Let p be an odd prime. For all $\delta \in (0,1)$ and for all $j \in \mathbb{F}_p^*$,*
(a) the matrices $M^{(j)}$ for learning linear functions over $\mathbb{F}_p^n$ satisfy $\tau_{M^{(j)}}(\delta) \leqslant -\frac{1-\delta}{2}$,
(b) the matrices $M^{(j)}$ for learning affine functions over $\mathbb{F}_p^n$ satisfy $\tau_{M^{(j)}}(\delta) \leqslant -\frac{1-\delta}{2} + \frac{\delta}{2n}$,
(c) the matrices $M^{(j)}$ for learning quadratic functions over $\mathbb{F}_p^n$ satisfy $\tau_{M^{(j)}}(\delta) \leqslant \frac{-(1-\delta)}{4} + \frac{2}{n}$, and
(d) for any $0 < \zeta < 1/2$, there are δ, γ with $0 < \delta < 1/2$ and $\gamma > 0$ such that for $d \leqslant \zeta n$, the matrices $M^{(j)}$ for learning functions of degree $\leqslant d$ over $\mathbb{F}_p^n$ satisfy $\tau_{M^{(j)}}(\delta) \leqslant -\gamma/d$.

Parts (a) and (b) of this theorem are immediate from matrix norm bounds. The proof of part (c) involves a tight structural characterization of quadratic polynomials over $\mathbb{F}_p$ and is in the full paper. The proof of part (d) for $d \geqslant 3$ uses tail bounds on the bias of polynomials of degree at most d over $\mathbb{F}_p$ recently proved by the authors in a companion paper (Beame et al., 2018b).

Using the above bounds on the norm amplification curves together with Theorem 4 we immediately obtain the time-space tradeoff lower bounds in following theorem.

Theorem 8 *Let p be an odd prime. There is an $\varepsilon > 0$ such that the following hold:*
(a) Any algorithm for learning linear or affine functions over $\mathbb{F}_p^n$ from their evaluations that succeeds with probability at least $p^{-\varepsilon n}$ requires time $p^{\Omega(n)}$ or space $\Omega(n^2 \log p)$.
(b) Let $m = \binom{n+2}{2}$. Any algorithm for learning quadratic functions over $\mathbb{F}_p^n$ that succeeds with probability at least $p^{-\varepsilon n}$ requires space $\Omega(nm \log p)$ or time $p^{\Omega(n)}$.
(c) There are constants $\zeta, \varepsilon > 0$ such that for $3 \leqslant d \leqslant (1-\zeta) \cdot n$ and for m equal to the number of monomials of degree at most d over $\mathbb{F}_p^n$, any algorithm for learning polynomial functions of degree at most d over $\mathbb{F}_p^n$ that succeeds with probability at least $p^{-\varepsilon n/d}$ requires space $\Omega(\log p \cdot nm/d)$ or time $p^{\Omega(n/d)}$.

5. Lower Bound for Learning Finite Functions from Samples

The full proof of Theorem 4 is given in the full paper. We sketch the main ideas here. For the $0 < \delta' < 1$ given in the theorem we define the significance threshold parameter $\delta = \delta'/6$, the bias threshold to $|A|^{-\gamma}$ for $\gamma = \min_j(-\tau_{M^{(j)}}(\delta')/2) > 0$, and consider any branching program B of length $< |A|^{\beta}$ and success probability at least $|A|^{-\beta/2}$ for $\beta = \min(\delta, \gamma)/8$. We prove that there must be at least $|X|^{\Omega(\delta\gamma \log_r |A|)} = |A|^{\Omega(\delta\gamma \log_r |X|)}$ significant nodes in B. We do so by proving that
(1) truncated paths reach significant vertices with probability at least $|A|^{-O(\beta)}$ but
(2) for any particular significant vertex s in B, the probability that a random truncated computation path taken by a random concept x ends at s with probability $|X|^{-\Omega(\delta\gamma \log_r |A|)}$.

The proof of (1) is easy and relies on the fact that prior to reaching a significant vertex, $\|\mathbb{P}_{x|v}\|_2 < |X|^{-(1-\delta)/2}$ and $\tau_{M^{(j)}}(\delta) \leqslant -2\gamma < 0$, so $\|M^{(j)} \cdot \mathbb{P}_{x|v}\|_2$ is small and hence the bias for a random input a is overwhelming likely to be smaller than $|A|^{-\gamma}$.

(2) follows according to a delicate progress argument where we measure the progress towards s at a vertex v (or by following an edge e) as

$$\rho(v) = \frac{\langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s}\rangle}{\langle \mathbb{P}_{x|s}, \mathbb{P}_{x|s}\rangle}, \qquad \text{respectively } \rho(e) = \frac{\langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s}\rangle}{\langle \mathbb{P}_{x|s}, \mathbb{P}_{x|s}\rangle}. \tag{1}$$

Clearly $\rho(s) = 1$ and we can show that the start vertex v_0 of B has $\rho(v_0) \leqslant |X|^{-\delta}$. We denote the set of vertices of B in the t-th layer by V_t. We overload this notation by identifying it with a probability distribution that gives each vertex $v \in V_t$ the probability that the random truncated computation path reaches v in layer t or $\perp$ if it never reaches the t-th layer. The progress at layer t is then measured as

$$\Phi_t = \mathbf{E}_{v\sim V_t}[(\rho(v))^{\gamma \log_r |A|}] \tag{2}$$

where we extend ρ to define $\rho(\perp) = 0$. By definition $\Phi_0 = |X|^{-\delta\gamma \log_r |A|}$. Since any path that reaches (and therefore ends at) s will contribute ρ value 1, we obtain the probability bound by showing that a high moment of ρ for random $v \sim V_t$ grows slowly with t and so is still extremely small. More precisely, for every t with $1 \leqslant t \leqslant |A|^{\beta} - 1$, we show

$$\Phi_t \leqslant \Phi_{t-1} \cdot (1 + |A|^{-2\beta}) + |X|^{-\gamma \log_r |A|}, \tag{3}$$

which yields the claimed lower bound. The argument requires an intermediate notion of progress in transferring from vertices to edges and back to vertices. If E_t is the set of edges from V_{t-1} to V_t (together with its overloaded probability distribution), we can define an analogous potential $\Phi'_t = \mathbf{E}_{e\sim E_t}[(\rho(e))^{\gamma \log_r |A|}]$. Merging edges into vertices loses information and we easily obtain $\Phi_t \leqslant \Phi'_t$ and it remains to measure the growth from vertices Φ_t to edges Φ'_{t+1}, which of course is where new information is learned. This follows from the key progress lemma which shows that for $v \in V_{t-1}$,

$$\sum_{e\in\Gamma_{out}(v)} \mathbf{Pr}[e|v] \cdot \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s}\rangle^{\gamma \log_r |A|} \leqslant (1 + |A|^{-2\beta}) \cdot \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s}\rangle^{\gamma \log_r |A|} + |X|^{-\gamma \log_r |A|}. \tag{4}$$

The second term in the sum covers the case when $\langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s}\rangle$ shows correlation worse than the uniform distribution, which could be improved dramatically in trivial ways and it is the $(1 + |A|^{-2\beta})$ multiplicative factor bound that is the important case. In this case, we show that up to a factor biased by a small amount away from 1, if edge e is associated with test a then $\langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s}\rangle$ is essentially a factor $(1 + \sum_{j=1}^{r-1} |(M^{(j)} \cdot \mathbb{P}_f)(a)|)$ larger than $\langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s}\rangle$ where $\mathbb{P}_f$ is a distribution that (essentially) gives points x' probability mass given by their proportional contribution of $\mathbb{P}_{x|v}(x')\mathbb{P}_{x|s}(x')$ to $\langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s}\rangle$. One can show that $\|\mathbb{P}_f\|_2$ itself is small relative to $\|\mathbb{P}_{x|s}\|_2$, which can't yet be too large since s is barely significant, and so again we use the fact that the norm amplification $\tau_{M^{(j)}}(\delta)$ is small and deduce that $\|M^{(j)} \cdot \mathbb{P}_f\|_2$ is tiny, which implies that each $|(M^{(j)} \cdot \mathbb{P}_f)(a)|$ is almost surely tiny. Plugging these in to the formula for the increase in the expected correlation overall all edges $e \in \Gamma_{out}(v)$ and analyzing the expectation of its higher power yields (4) and hence the bound on Φ'_{t+1} and therefore Theorem 4.

6. An SDP Relaxation for Norm Amplification on the Positive Orthant

For a matrix $M \in \mathbb{C}^{A\times X}$, $\tau_M(\delta) = \sup_{\substack{\mathbb{P}\in\Delta_X \\ \|\mathbb{P}\|_2 \leqslant 1/|X|^{1-\delta/2}}} \log_{|A|}(\|M\cdot\mathbb{P}\|_2)$.

That is, $\tau_M(\delta) = \frac{1}{2}\log_{|A|} OPT_{M,\delta}$ where $OPT_{M,\delta}$ is the optimum of the following quadratic program:

$$\begin{aligned} \text{Maximize} \quad & \|M\cdot\mathbb{P}\|_2^2 = \langle M\cdot\mathbb{P}, M\cdot\mathbb{P}\rangle, \\ \text{subject to:} \quad & \sum_{i\in X}\mathbb{P}_i = 1, \\ & \sum_{i\in X}\mathbb{P}_i^2 \leqslant |X|^{\delta-1}, \\ & \mathbb{P}_i \geqslant 0 \qquad\qquad \text{for all } i\in X. \end{aligned} \tag{5}$$

Instead of attempting to solve (5), presumably a difficult quadratic program, we consider the following semidefinite program (SDP):

$$\begin{aligned} \text{Maximize} \quad & \langle M^*M, U\rangle\cdot|X|^2/|A| \\ \text{subject to:} \quad & \\ [V] \quad & U \succeq 0, \\ [w] \quad & \sum_{i,j\in X} U_{ij} = 1, \\ [z] \quad & \sum_{i\in X} U_{ii} \leqslant |X|^{\delta-1}, \\ & U_{ij}\in\mathbb{R}, U_{ij}\geqslant 0 \qquad \text{for all } i,j\in X. \end{aligned} \tag{6}$$

Recall that M^* is the conjugate transpose of M. Note that for any $\mathbb{P}\in\Delta_X$ achieving the optimum value of (5) the positive semidefinite matrix $U = \mathbb{P}\cdot\mathbb{P}^T$ has the same value in (6) (where the $|X|^2/|A|$ factor accounts for the difference in scaling factors based on the dimensions for the two expectation inner products), and hence (6) is an SDP relaxation of (5).

But this is not a standard SDP, since M is over $\mathbb{C}$ and M^*M might contain complex entries. In order to apply techniques on real matrices, we define $N : X\times X \to \mathbb{R}$ as $N(x,x') = Re(M^*M(x,x'))$, that is, N is the real part of M^*M. Then we obtain a (real) definite program that is identical to (6), except that M^*M is replaced by N and the condition $U_{ij}\in\mathbb{R}$ is superfluous. The key observation is that (6) and this real program (not shown) have the same optimal value. This is because for any $U\in\mathbb{R}^{X\times X}$,

$$|X|^2\langle M^*M, U\rangle = \sum_{i,j}(M^*M)_{ij}\cdot U_{ij} = \sum_{ij} Re((M^*M)_{ij})\cdot U_{ij} + i\cdot\sum_{x,x'} Im((M^*M)_{ij})\cdot U_{ij}$$

Since M^*M is a Hermitian matrix, we have $(M^*M)_{ij} = \overline{(M^*M)_{ji}}$. But U is real symmetric, so we have $\sum_{i,j} Im((M^*M)_{ij})\cdot U_{ij} = 0$, namely

$$|X|^2\langle M^*M, U\rangle = \sum_{i,j} Re((M^*M)_{ij})\cdot U_{ij} = |X|^2\langle N, U\rangle$$

and we only need to consider the real parts.

In order to upper bound the value of (6), we consider the dual program to this real SDP which can be written as:

$$\begin{aligned} \text{Minimize} \quad & w + z \cdot |X|^{\delta} \cdot |X|^{-1} \\ \text{subject to:} \quad & V \succeq 0, \\ & zI + wJ \geqslant V + N/|A|, \\ & z \geqslant 0. \end{aligned} \tag{7}$$

where I is the identity matrix and J is the all 1's matrix over $X \times X$.

Any dual solution of (7) yields an upper bound on the optimum of (6) and hence $OPT_{M,\delta}$ and $\tau_M(\delta)$. To simplify the complexity of analysis we restrict ourselves to considering semidefinite matrices V that are suitably chosen Laplacian matrices. For any set S in $X \times X$ and any $\alpha : S \to \mathbb{R}_+$ the Laplacian matrix associated with S and α is defined by $L_{(S,\alpha)} := \sum_{(i,j)\in S} \alpha(i,j) L_{ij}$ where $L_{ij} = (e_i - e_j)(e_i - e_j)^T$ for the standard basis $\{e_i\}_{i\in X}$. Intuitively, in the dual SDP (7), by adding matrix $V = L_{S,\alpha}$ for suitable S and α depending on M we can shift weight from the off-diagonal entries of N to the diagonal where they can be covered by the $z + w$ entries on the diagonal rather than being covered by the w values in the off-diagonal entries. This will be advantageous for us since the objective function has much smaller coefficient for z which helps cover the diagonal entries than coefficient for w, which is all that covers the off-diagonal entries.

Definition 9 *Suppose that $N \in \mathbb{R}^{X\times X}$ is a symmetric matrix. For $\kappa \in \mathbb{R}_+$, define*

$$W_\kappa(N) = \max_{i\in X} \sum_{j\in X:\ N_{i,j}>\kappa} (N_{i,j} - \kappa).$$

The following lemma is the basis for our bounds on $\tau_M(\delta)$.

Lemma 10 *Let $\kappa \in \mathbb{R}_+$. Then $OPT_{M,\delta} \leqslant (\kappa + W_\kappa(N) \cdot |X|^{\delta-1})/|A|$.*

Proof For each off-diagonal entry of N with $N(i,j) > \kappa$, include matrix L_{ij} with coefficient $(N(i,j) - \kappa)/|A|$ in the sum for the Laplacian V. By construction, the matrix $V + N/|A|$ has off-diagonal entries at most $\kappa/|A|$ and diagonal entries at most $(\kappa + W_\kappa(N))/|A|$. The solution to (7) with $w = \kappa/|A|$ and $z = W_\kappa(N)/|A|$ is therefore feasible, which yields the bound as required. ■

It may not be easy to bound $W_\kappa(N)$ directly, since the real part of M^*M may not have good structure. Fortunately, we have the following measure:

Definition 11 *Let $M \in \mathbb{C}^{A\times X}$ be a complex matrix. For $\kappa \in \mathbb{R}_+$, define*

$$\tilde{W}_\kappa(M) = \max_{i\in X} \sum_{j\in X:\ |(M^*M)_{i,j}|>\kappa} (|(M^*M)_{i,j}| - \kappa).$$

Proposition 12 *Let $\kappa \in \mathbb{R}_+$. Then $W_\kappa(N) \leqslant \tilde{W}_\kappa(M)$*

Proof Whenever $N_{i,j} > \kappa$, we have $|(M^*M)_{i,j}| \geqslant N_{i,j} > \kappa$. Moreover, this gives $|(M^*M)_{i,j}| - \kappa \geqslant N_{i,j} - \kappa$. Then the statement follows the two definitions. ■

For specific matrices M, we obtain the required bounds on $\tau_M(\delta) < 0$ for some $0 < \delta < 1$ by showing that we can set $\kappa = |A|^\gamma$ for some $\gamma < 1$ and obtain that $W_\kappa(N)$ or $\tilde{W}_\kappa(M)$ is at most $\kappa \cdot |X|^{\gamma'}$ for some $\gamma' < 1$.

References

Paul Beame, Shayan Oveis Gharan, and Xin Yang. Time-space tradeoffs for learning from small test spaces: Learning low degree polynomial functions. Technical Report TR17-120, Electronic Colloquium on Computational Complexity (ECCC), 2017.

Paul Beame, Shayan Oveis Gharan, and Xin Yang. Time-space tradeoffs for learning finite functions from random evaluations, with applications to polynomials. Technical Report TR18-114, Electronic Colloquium on Computational Complexity (ECCC), 2018a.

Paul Beame, Shayan Oveis Gharan, and Xin Yang. On the bias of Reed-Muller codes over odd prime fields. *arXiv preprint*, 2018b.

Ido Ben-Eliezer, Rani Hod, and Shachar Lovett. Random low-degree polynomials are hard to approximate. *Computational Complexity*, 21(1):63–81, 2012. URL https://doi.org/10.1007/s00037-011-0020-6.

Sumegha Garg, Ran Raz, and Avishay Tal. Extractor-based time-space tradeoffs for learning. Technical Report TR17-121, Electronic Colloquium on Computational Complexity (ECCC), 2017.

Sumegha Garg, Ran Raz, and Avishay Tal. Extractor-based time-space lower bounds for learning. In *Proceedings of the Fiftieth Annual ACM on Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA*, 2018. To appear.

Gillat Kol, Ran Raz, and Avishay Tal. Time-space hardness of learning sparse parities. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1067–1080, 2017. URL http://doi.acm.org/10.1145/3055399.3055430.

Robert James McEliece. *Linear recurring sequences over finite fields*. PhD thesis, California Institute of Technology, 1967.

Dana Moshkovitz and Michal Moshkovitz. Mixing implies lower bounds for space bounded learning. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, pages 1516–1566, 2017. URL http://proceedings.mlr.press/v65/moshkovitz17a.html.

Dana Moshkovitz and Michal Moshkovitz. Entropy samplers and strong generic lower bounds for space bounded learning. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 28:1–28:20, 2018. URL https://doi.org/10.4230/LIPIcs.ITCS.2018.28.

Ran Raz. Fast learning requires good memory: A time-space lower bound for parity learning. In *Proceedings, 57th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2016, New Brunswick, New Jersey, USA*, pages 266–275, October 2016. doi: 10.1109/FOCS.2016.36. URL http://dx.doi.org/10.1109/FOCS.2016.36.

Ran Raz. A time-space lower bound for a large class of learning problems. In *Proceedings, 58th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, California, USA*, pages 732–742, October 2017.

Ohad Shamir. Fundamental limits of online and distributed algorithms for statistical learning and estimation. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 163–171, Montreal, Quebec, Canada, 2014. URL https://arxiv.org/pdf/1311.3494.pdf.

Neil J. A. Sloane and Elwyn R. Berlekamp. Weight enumerator for second-order Reed-Muller codes. *IEEE Trans. Information Theory*, 16(6):745–751, 1970. doi: 10.1109/TIT.1970.1054553.

Jacob Steinhardt, Gregory Valiant, and Stefan Wager. Memory, communication, and statistical queries. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA*, pages 1490–1516, 2016. URL http://jmlr.org/proceedings/papers/v49/steinhardt16.html.