Towards a Combinatorial Characterization of Bounded-Memory Learning

Alon Gonen **Shachar Lovett** **Michal Moshkovitz**

University of California San Diego

Abstract

Combinatorial dimensions play an important role in the theory of machine learning. For example, VC dimension characterizes PAC learning, SQ dimension characterizes weak learning with statistical queries, and Littlestone dimension characterizes online learning. In this paper we aim to develop combinatorial dimensions that characterize bounded memory learning. We propose a candidate solution for the case of realizable strong learning under a known distribution, based on the SQ dimension of neighboring distributions. We prove both upper and lower bounds for our candidate solution, that match in some regime of parameters. This is the first characterization of strong learning under space constraints in any regime. In this parameter regime there is an equivalence between bounded memory and SQ learning. We conjecture that our characterization holds in a much wider regime of parameters.

1 Introduction

Characterization of different learning tasks using a combinatorial condition has been investigated in depth in machine learning. Learning a class in an unconstrained fashion is characterized by a finite VC dimension [40, 9], and weakly learning in the statistical query (SQ) framework is characterized by a small SQ dimension [7]. Is there a simple combinatorial condition that characterizes learnability with bounded memory? In this paper we propose a candidate condition, prove upper and lower bounds that match in some of the regime of parameters, and conjecture that they match in a much wider regime of parameters.

A learning algorithm that uses b bits of memory, m samples, and accuracy $1-\epsilon$ is defined as follows: the algorithm receives a series of m labeled examples one by one, while only preserving an internal state in $\{0,1\}^b$ between examples. In this paper we focus our attention on the realizable setting: the labeled examples are pairs $(x_i, c(x_i))$, where $x_i \in \mathcal{X}$ and $c : \mathcal{X} \to \{-1, 1\}$ is a concept in a concept class $\mathcal{C}$. The algorithm is supposed to return with constant probability a hypothesis h which matches the unknown concept c on a $1-\epsilon$ fraction of the underlying distribution. In this paper we further assume that the underlying distribution P on $\mathcal{X}$ is known to the learner, similar to the setting in the SQ framework.

There are two "trivial" algorithms for the problem which we now present. For ease of presentation, we restrict our attention in the introduction to a small constant ϵ, say $\epsilon = 0.01$. Without making any additional assumptions, the following space complexity bounds are known when learning with accuracy 0.99:

1. The ERM algorithm keeps in memory $m = O(\log |\mathcal{C}|)$ samples, and outputs a hypothesis that is consistent with the entire sample. This requires $b = O(\log |\mathcal{C}| \log |\mathcal{X}|)$ bits.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

2. A learning algorithm that enumerates all possible concepts in $\mathcal{C}$ and the consistency of each concept based on few random samples. This algorithms requires $m = O(|\mathcal{C}| \log |\mathcal{C}|)$ samples and $b = O(\log |\mathcal{C}|)$ bits.

We define a class $\mathcal{C}$ under a distribution P to be *learnable with bounded memory* if there is a "non-trivial" learning algorithm with respect to both sample complexity and space complexity. A bit more formally, if there is a learning algorithm that uses only $m = |\mathcal{C}|^{o(1)}$ samples and $b = o(\log |\mathcal{C}| \log |\mathcal{X}|)$ bits (see Definition 1).

A crucial combinatorial measure that has been linked to bound-memory *weak* learning is the *statistical query* (SQ) dimension (see Definition 3). Extending these results to *strong* learning requires the following definition. We say that a distribution Q is μ-close to the distribution P (where $\mu \geq 1$) if the ratio $P(x)/Q(x)$ is between $1/\mu$ and μ for all points x in the domain. We denote by $\mathcal{P}_\mu(P)$ the set of all distributions which are μ-close to P (see Definition 2).

Our main results are upper and lower bounds on bounded memory learning, in terms of the SQ dimension of distributions in the neighbourhood of the underlying distribution P. While deriving tighter bounds that hold in a wider regime remains an important open question, these are the first characterizations of the space complexity of *strong* learning using the SQ dimension.

1. Suppose that there is a parameter $d \geq 1$ such that for any distribution $Q \in \mathcal{P}_d(P)$ it holds that $\mathrm{SQ}_Q(\mathcal{C}) \leq d$. Then there exists an algorithm that learns the class $\mathcal{C}$ with accuracy 0.99 under the distribution P using $b = O(\log(d) \cdot \log |\mathcal{C}|)$ bits and $m = \mathrm{poly}(d) \cdot \log(|\mathcal{C}|) \cdot \log\log(|\mathcal{C}|)$ samples.
2. If the class $\mathcal{C}$ is PAC-learnable under P with accuracy 0.99 using b bits and m samples, then for every distribution $Q \in \mathcal{P}_{\Theta(1)}(P)$ its SQ dimension is bounded by $SQ_Q(\mathcal{C}) \leq \max(\mathrm{poly}(m), 2^{O(\sqrt{b})})$.

In Section 1.2 we give a more detailed account of the bounds for general ϵ. We show that for small enough ϵ, the two conditions coincide and we in fact get a characterization of bounded memory learning. We conjecture that the characterization holds for a larger range of parameters (see Conjecture 1). We also prove similar conditions for SQ learning, thus implying equivalence between bounded memory learning and SQ learning for small enough ϵ.

1.1 Problem setting

In this paper we consider two learning frameworks: a) The PAC model [39] and b) The Statistical Query framework [21]. See a recap of these frameworks in Appendix A.

Bounded memory learning. A bounded memory learning algorithm observes a sequence of labeled examples $(x_1, y_1), (x_2, y_2), \ldots$ in a streaming fashion, where $x_i \in \mathcal{X}, y_i \in \{-1, 1\}$. We assume in this paper that the data is realizable, namely $y_i = c(x_i)$ for some concept $c \in \mathcal{C}$. The algorithm maintains a state $Z_t \in \{0,1\}^b$ after seeing the first t examples, and updates it after seeing the next example to $Z_{t+1} = \psi_t(Z_t, (x_{t+1}, y_{t+1}))$ using some update function ψ_t.[1] The parameter b is called the *bit complexity* of the algorithm. Finally, after observing m samples (where m is a parameter tuned by the algorithm), a hypothesis $h = \phi(Z_m)$ is returned.

We now expand the "trivial" learning algorithms described earlier to accuracy $1 - \epsilon$ for any $\epsilon > 0$:

1. We can learn with accuracy $1 - \epsilon$ using $m = O\left(\log |\mathcal{C}| \mathrm{poly}(1/\epsilon)\right)$ samples and number of bits equal to $b = O\left(\log |\mathcal{C}| \log |\mathcal{X}| + \log |\mathcal{C}| \log(1/\epsilon)\right)$. For constant accuracy parameter this can be done by saving $O(\log |C|)$ examples and applying ERM. To achieve better accuracy we can apply Boosting-By-Majority [15] as we describe in Section 3.
2. One can always learn with $m = O(|\mathcal{C}| \log |\mathcal{C}| \epsilon^{-1})$ samples and $b = O(\log |\mathcal{C}|)$ bits, by going over all possible hypothesis and testing if the current hypothesis is accurate on a few random samples.

[1] Following the model of branching programs (e.g., [29]), the maps $\psi_1, \psi_2, \ldots$ are not considered towards the space complexity of the algorithm.

We define a class $\mathcal{C}$ to be bounded memory learnable if there is a learning algorithm that beats both of the above learning algorithms. Bounded-memory algorithms should be allowed to save at least a hypothesis and an example in memory. But in extreme cases saving one hypothesis means allowing saving the entire training data in memory. Thus, the definition is most appropriate for the case that $|\mathcal{C}|$ is about the same as $|\mathcal{X}|$.

Definition 1 (Bounded memory learnable classes)**.** *A class $\mathcal{C}$ under a distribution P is* learnable with bounded memory *with accuracy $1-\epsilon$ if there is a learning algorithm that uses only $m = (|\mathcal{C}|/\epsilon)^{o(1)}$ samples and $b = o(\log|\mathcal{C}|(\log|\mathcal{X}| + \log(1/\epsilon)))$ bits*[2].

To illustrate this, consider the case where the number of concepts and points are polynomially related, $|\mathcal{C}|, |\mathcal{X}| = \text{poly}(N)$, and where the desired error in not too tiny, $\epsilon \geq 1/\text{poly}(N)$. Then a non-trivial learning algorithm is one that uses a sub-polynomial number of samples $m = N^{o(1)}$ and a sub-quadratic number of bits $b = o(\log^2 N)$. There are classes that can not be learned with bounded memory.

Example 1 (Learning parities)**.** *Consider the task of learning parities on n bits. Concretely, let $N = 2^n$, $\mathcal{X} = \mathcal{C} = \{0,1\}^n$, P be the uniform distribution over $\mathcal{X}$, and let the label associated with a concept $c \in \mathcal{C}$ and point $x \in \mathcal{X}$ be $\langle c, x\rangle \pmod 2$. It was shown by [29, 26] that achieving constant accuracy for this task requires either $b = \Omega(n^2) = \Omega(\log^2 N)$ bits of memory or an exponential in n many samples, namely $m = 2^{\Omega(n)} = N^{\Omega(1)}$ samples.*

Close distributions. An important ingredient in this work is the notion of nearby distributions, where the distance is measured by the multiplicative gap between the probabilities of elements.

Definition 2 (μ-close distributions)**.** *We say that two distributions P, Q on $\mathcal{X}$ are μ-close for some $\mu \geq 1$ if $\mu^{-1}P(x) \leq Q(x) \leq \mu P(x)$ for all $x \in \mathcal{X}$. Note that the definition is symmetric with respect to P, Q. We denote the set of all distributions that are μ-close to P by $\mathcal{P}_\mu(P)$.*

1.2 Main results

Bounded memory PAC learning. We state our main results for a combinatorial characterization of bounded memory PAC learning in terms of the SQ dimension of distributions close to the underlying distribution.

Theorem 1. *Let $\epsilon \in (0,1)$, $d \in \mathbb{N}$ and denote by $\mu = \Theta(\max\{d, 1/\epsilon^3\})$. Suppose that the distribution P satisfies the following condition: for any distribution $Q \in \mathcal{P}_\mu(P)$, $\text{SQ}_Q(\mathcal{C}) \leq d$. Then there exists an algorithm that learns the class $\mathcal{C}$ with accuracy $1-\epsilon$ under the distribution P using $b = O(\log(d/\epsilon) \cdot \log|\mathcal{C}|)$ bits and $m = \text{poly}(d/\epsilon) \cdot \log(|\mathcal{C}|) \cdot \log\log(|\mathcal{C}|)$ samples.*

Theorem 2. *If a class $\mathcal{C}$ is strongly PAC-learnable under P with accuracy $1 - 0.1\epsilon$ using b bits and m samples, then for every distribution $Q \in \mathcal{P}_{1/\epsilon}(P)$, its SQ-dimension is bounded by $SQ_Q(\mathcal{C}) \leq \max\left(\text{poly}(m/\epsilon), 2^{O(\sqrt{b})}\right)$.*

There is a regime of parameters where the upper and lower bounds match. Let $|\mathcal{C}|, |\mathcal{X}| = \text{poly}(N)$ and that $\epsilon = N^{-o(1)}$. Recall that the class is bounded memory learnable if there is a learning algorithm with sample complexity $m = N^{o(1)}$ and space complexity $b = o(\log^2 N)$. Let $\mu, d = N^{o(1)}$. We have the following equivalence, which we conjecture holds for any ϵ:

$$\begin{array}{c} \mathcal{C} \textit{ is bounded memory learnable under } P \textit{ with accuracy } 1-\epsilon \\ \Updownarrow \\ \forall Q \in \mathcal{P}_{poly(1/\epsilon)}(P),\ \text{SQ}_Q(\mathcal{C}) \leq poly(1/\epsilon)\ . \end{array}$$

Conjecture 1. *For any ϵ, the class $\mathcal{C}$ is bounded memory learnable under distribution P with accuracy $1-\epsilon \iff \forall Q \in \mathcal{P}_{poly(1/\epsilon)}(P)$, $\text{SQ}_Q(\mathcal{C}) \leq poly(1/\epsilon)$.*

SQ learning. Next, we give our secondary results for SQ learning, which are very similar to our results for bounded memory learning. Conceptually, it shows that the two notions are tightly connected.

[2]Formally, the $o(\cdot)$ factors are in terms of the size of the class $\mathcal{C}$. Hence this definition applies to families of distributions $\{\mathcal{C}_n\}$ of growing size, for example parities on n bits. However, in the main theorems we give quantitative bounds and hence can focus on single classes instead of families of classes.

Theorem 3. *Let $\epsilon \in (0,1)$, $d \in \mathbb{N}$ and denote by $\mu = \Theta(\max\{d, 1/\epsilon^3\})$. Suppose that the distribution P satisfies the following condition: for any distribution $Q \in \mathcal{P}_\mu(P)$, $\mathrm{SQ}_Q(\mathcal{C}) \leq d$. Then there exists an SQ-learner that learns the class $\mathcal{C}$ with accuracy $1-\epsilon$ under the distribution P using $q = \mathrm{poly}(d/\epsilon)$ statistical queries with tolerance $\tau \geq \mathrm{poly}(\epsilon/d)$.*

Theorem 4. *If a class $\mathcal{C}$ is strongly SQ-learnable under P with accuracy $1-0.1\epsilon$, q statistical queries, and tolerance τ, then for every distribution $Q \in \mathcal{P}_{1/\epsilon}(P)$, $SQ_Q(\mathcal{C}) \leq \mathrm{poly}(q/\epsilon\tau)$.*

Note that for any class $\mathcal{C}$, underlying distribution and accuracy $1-\epsilon$, one can SQ-learn the class with $q = |\mathcal{C}|$ statistical queries and tolerance $\tau = O(\epsilon)$, by going over all the hypotheses. Thus a class is non-trivially SQ-learnable if one can learn it with $q = |C|^{o(1)}$ queries and tolerance $\tau \geq \mathrm{poly}(\epsilon)$. Focusing on the case that $|\mathcal{C}|, |\mathcal{X}| = \mathrm{poly}(N)$ and $\mu, d, q, 1/\epsilon, 1/\tau = N^{o(1)}$, we get that bounded memory learning is equivalent to SQ learning.

1.3 Related work

Characterization of bounded memory learning. Many works have proved lower bounds under memory constraints [34, 29, 23, 25, 26, 30, 17, 12, 4, 35, 18, 11]. Some of these works even provide a necessary condition for learnability with bounded memory. As for upper bounds, not many works have tried to give a general property that implies learnability under memory constraints. One work suggested such property [27] but this did not lead to a full characterization of bounded memory learning.

Statistical query learning. After Kearns's introduction of statistical query [21], Blum et al. [7] characterized weak learnability using SQ dimension. Specifically, if $SQ_P(\mathcal{C}) = d$, then $\mathrm{poly}(d)$ queries are both needed and sufficient to learn with accuracy $1/2 + \mathrm{poly}(1/d)$. Note that the advantage is very small, only $\mathrm{poly}(1/d)$. Subsequently several works [3, 36, 38, 13] suggested a few characterizations of strong SQ learnability.

Bounded memory and SQ dimension. In this paper we prove an equivalence, in some parameters regime, between bounded memory learning and SQ learning. There were a few indications in the literature that such an equivalence exists. The work [37] showed a general reduction from any SQ learner to a memory efficient learner. Alas, they gave an example that suggests that an equivalence is incorrect, which we now address.

Example 2 (Learning sparse parity)**.** *Consider the concept class of parity on the first k bits of an n-bit input for $k \ll n$, for example $k = \sqrt{n}$. That is, $\mathcal{X} = \{0,1\}^n$ and $\mathcal{C} = \{0,1\}^k \cdot \{0\}^{n-k}$ is a subset of all possible parities. Naively, an ERM algorithm would need to store $\Theta(k)$ examples, each requiring n bits, and hence need $b = \Theta(kn)$. However, it suffices to store only the first k bits of each example, and hence only use $b = \Theta(k^2)$ bits. As this is significantly less than the naive bound of $\Theta(kn)$ we consider the class to be bounded memory learnable. On the other hand, the SQ dimension of $\mathcal{C}$ is maximal, namely 2^k, and hence [37] suggest that this example separates bounded memory learning and SQ learning.*

Relating to our results, it shows two things: when the sizes of the concept classes $\mathcal{C}$ and example set $\mathcal{X}$ are polynomially related, there is no such separation (we prove this for small enough ϵ and conjecture for all ϵ). Moreover, the $2^{O(\sqrt{b})}$ term in Theorem 2 is tight.

The work [17] showed that high SQ dimension implies non-learnability with bounded memory when the learner returns the exact answer. However, learnability is usually inexact and this does not relate to strong learnability.

Littlestone dimension. Online learnability without memory constraints is characterized using Littlestone dimension [24]. This dimension is not suited for bounded memory learning as it does not take into account the structure of the class which determines whether the class is learnable with bounded memory or not. Specifically, there are problems that have similar Littlestone dimension (e.g., parity and discrete thresholds on the line), where the former (thresholds) is easy to learn under memory constraints and the latter (parity) is hard.

Learning under a known distribution. In SQ framework most works focused on learning under known distributions [7, 10, 43, 44, 3, 36, 13, 38]. However, PAC learning research under known

distribution is scarce but exists, e.g., [6, 5, 41, 31]. In particular, Benedek et al. [6] showed that unconstrained learning under known distribution is characterized by covering.

Smooth distributions. A key idea in this paper is to use *close* distributions which are upper and lower bounded by a distribution. A one sided closeness, namely the upper bound, is referred in the literature as a *smooth* distribution, see for example [10]. Smooth distributions were also used to show equivalence between boosting and hard-core sets [22, 19].

Paper organization. We begin in Section 2 with a presentation of known results in boosting and statistical queries that we will need. In Section 3 we construct learning algorithms based on the assumption that close distributions have bounded SQ dimensions, and prove Theorem 1 and Theorem 3. In Section 4 we establish the reverse direction and prove Theorem 2 and Theorem 4. Omitted proofs can be found in the appendix.

2 Preliminaries

Weak learning and boosting. It is often conceptually easier to design an algorithm whose accuracy is slightly better than an educated guess, and then attempt to boost its accuracy.

Consider first the PAC model. We say that a learning algorithm $\mathcal{W}$ is a γ-*weak learner* if there exists an integer m such that for any target concept $c \in \mathcal{C}$ and any $n \geq m$, with probability at least $2/3$ over the draw of an i.i.d. labeled sample $S = ((x_1, c(x_1)), \ldots, (x_n, c(x_n)))$ according to the underlying distribution P, the hypothesis returned by $\mathcal{A}$ is $(1/2 - \gamma)$-accurate. We refer to the minimal integer m satisfying the above as the *sample complexity* of the weak learner. The notion of γ-weak learning in the SQ framework is defined analogously, where the *query complexity* of the weak learner is denoted by q_τ (where τ is the tolerance parameter).

A *boosting* algorithm $\mathcal{A}$ uses an oracle access to a weak learner $\mathcal{W}$ and aggregates the predictions of $\mathcal{W}$ into a satisfactory accurate solution. The celebrated works of Freund and Schapire [32, 33, 14, 15, 16] provide several successful boosting algorithms for the PAC model. The work of [2] extended some of these results to the SQ framework.

Known SQ-dimension bounds for weak learning. The following upper and lower bounds are known. The first upper bound is a folklore lemma whose proof can be found in [38].

Proposition 1. *Let $\mathcal{C}$ be a concept class, P an underlying distribution, such that $\mathrm{SQ}_P(\mathcal{C}) \leq d$. Then there is a $(1/d)$-weak SQ-learner with query complexity $q = d$ and tolerance $\tau = 1/3d$.*

The next lower bound was initially proved by [8]. A simplified proof was given later by [38].

Proposition 2. *Let $\mathcal{C}$ be a concept class, P an underlying distribution, and let $d = SQ_P(\mathcal{C})$. Any learning algorithm that uses tolerance parameter lower bounded by $\tau > 0$ requires in the worst case at least $(d\tau^2 - 1)/2$ queries for learning $\mathcal{C}$ with accuracy at least $1/2 + 1/d$.*

Finally, the next proposition shows that SQ learnability (weak or strong) implies learning with bounded memory.

Proposition 3 (Theorem 7 in [37])**.** *Assume that a class $\mathcal{C}$ can be learned using q statistical queries with tolerance τ. Then there is an algorithm that learns $\mathcal{C}$ using $m = O(\frac{q \log |\mathcal{C}|}{\tau^2}(\log(q) + \log\log(|\mathcal{C}|))$ samples and $b = O(\log |\mathcal{C}| \cdot \log(q/\tau))$ bits.*

3 From bounded SQ dimension to bounded memory learning

In this section we prove our upper bounds: Theorem 1 and Theorem 3. A schematic illustration of the proof is given in Fig. 1.

Overview. To prove Theorem 1 we apply an extension of the Boosting-By-Majority (BBM) algorithm [15] to the SQ framework due to [2]. Similarly to other popular boosting methods (e.g. AdaBoost [16]), the algorithm operates by re-weighting the input sample and feeding the weak learner with sub-samples drawn according to the re-weighed distributions. The main challenge is to bound

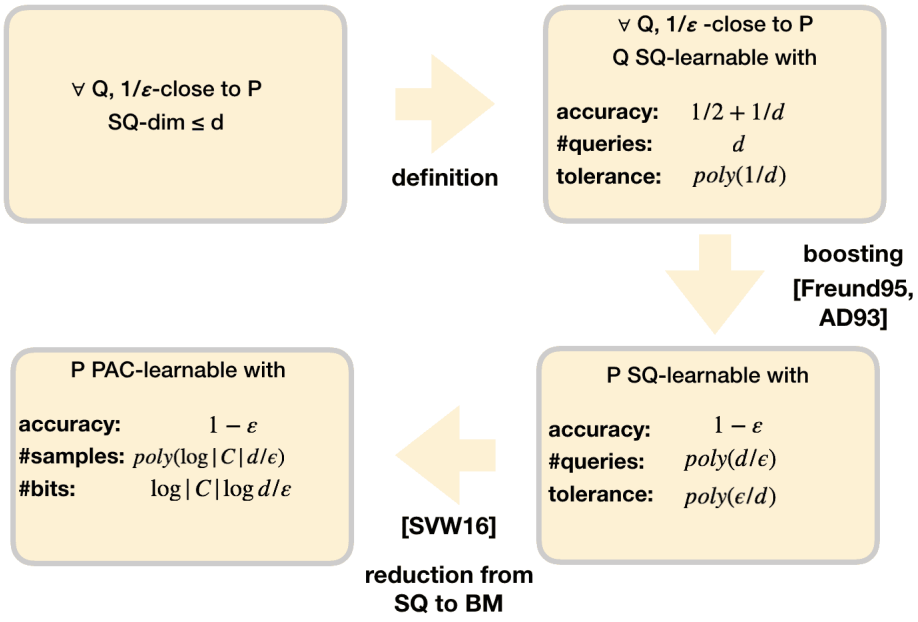

Figure 1: Proof outline (with asymptotic terms): from bounded SQ dimension under close distributions to strong learnability.

the SQ-dimension of the probability distributions maintained by the boosting algorithm. This will allow us to obtain a bound on the query complexity of the boosting process using Proposition 1 and thus conclude Theorem 1. Consequently, we deduce Theorem 3 using Proposition 3.

SQ-Boost-By-Majority. Following [2] we describe how BBM can be carried out in the SQ model. Instead of having an access to a sampling oracle, the booster $\mathcal{A}$ has an access to an SQ oracle with respect to the distribution P and the target concept c. Similarly to BBM, the booster re-weights the points in $\mathcal{X}$ in iterative fashion, thereby defining a sequence of distributions, $P_1, \ldots, P_T$. The weak learner $\mathcal{W}$ itself also works in the SQ model. That is, instead of requiring samples $S_1, \ldots, S_T$ drawn according to $P_1, \ldots, P_T$, it submits statistical queries to the boosting algorithm. The guarantee of the weak learner remains intact; provided that it gets sufficiently accurate answers (as determined by the tolerance parameter τ), $\mathcal{W}$ should output a weak classifier whose correlation with the target concept is at least γ.

Therefore, the challenging part in translating BBM to the SQ model is to enable simulating answers to statistical queries with respect to the distributions $P_1, \ldots, P_T$ given only an access to an SQ oracle with respect to the initial distribution P. Fortunately, the BBM's re-weighting scheme makes it rather easy. It follows from the definition of the distributions maintained by BBM (see Eq. (1) and Eq. (2)) that in the beginning of round t, the space $\mathcal{X}$ partitions into t regions such that the probability of points in each region is proportional to their initial distribution according to P. This allows simulating an *exact* SQ query with respect to P_{t+1} using $O(t)$ exact SQ queries to P. Furthermore, as shown in [2], the fact that $P_t(x) \leq (C/\epsilon^3) \cdot P(x)$ allows us to perform this simulation with suitable tolerance parameters. This is summarized in the next proposition.

Proposition 4 ([2]). *Any statistical query with respect to the distribution P_t with tolerance τ can be simulated using $O(t)$ statistical queries with respect to the original distribution P with tolerance parameter $\Omega(\tau \cdot \mathrm{poly}(\epsilon))$.*

Upper bounding the SQ-dimension of SQ-BBM's distributions. In this part we derive an upper bound on the SQ-dimension of the distribution $P_1, \ldots, P_T$ maintained by SQ-BBM. To this end we use our assumption that for all $Q \in \mathcal{P}_\mu(P)$, $\mathrm{SQ}_Q(\mathcal{C}) \leq d$ where $\mu = \max\{C/\epsilon^3, 4d\}$. While we

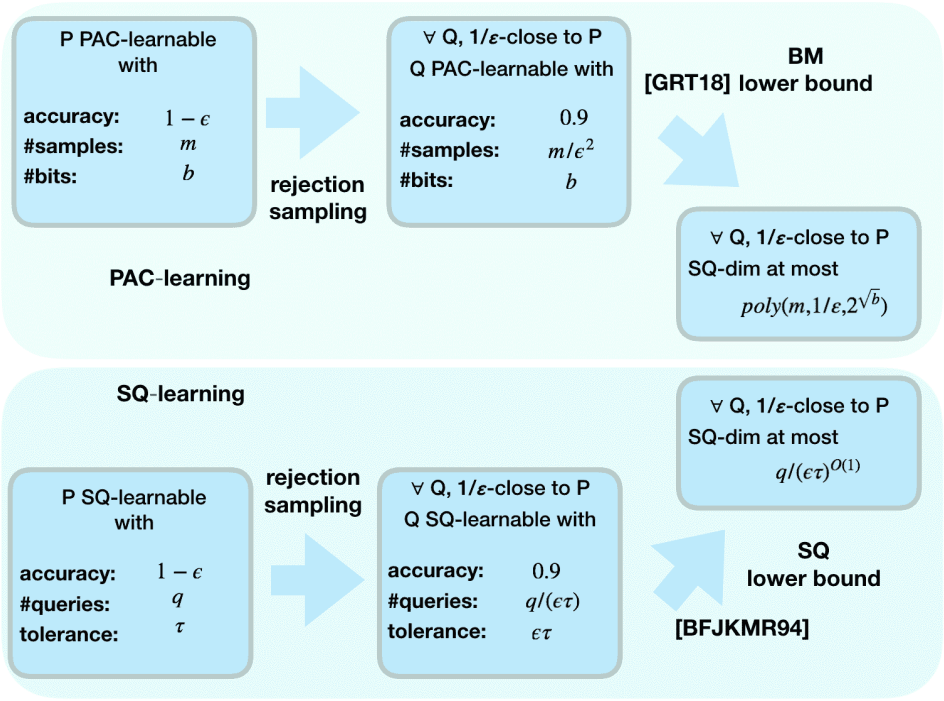

Figure 2: Proof outline (with asymptotic terms): from strong learnability to bounded SQ-dimension under close distributions in PAC and SQ models.

cannot make sure that the distributions $P_1, \ldots, P_T$ belong to $\mathcal{P}_\mu(P)$, we will still be able to derive an upper bound on their SQ-dimension.

Lemma 1. *Let $P_1, \ldots, P_T$ be the distributions maintained by SQ-BBM. For every $t = 1, \ldots T$, $\mathrm{SQ}_{P_t}(\mathcal{C}) \leq 4d$.*

Putting it all together. We now complete the proofs of Theorem 3 and Theorem 1.

Proof of Theorem 3. From Proposition 1 we conclude that for any $Q \in \mathcal{P}_\mu(P)$ there exists a $(1/d)$-weak learner with query complexity d and tolerance $1/(3d)$. Using this weak learner we apply SQ-BBM as described above. From Lemma 1 we know that for every distribution P_t maintained by SQ-BBM, $\mathrm{SQ}_{P_t}(\mathcal{C}) = O(d)$. Combining Proposition 6 and Proposition 4 we conclude that SQ-BBM reaches a $1-\epsilon$ accurate prediction after $T = O(\mathrm{poly}(d)\log(1/\epsilon))$ iterations while using at most $\mathrm{poly}(d/\epsilon)$ statistical queries with tolerance at least $\mathrm{poly}(\epsilon/d)$. □

Proof of Theorem 1. Proposition 3 tells us that if a class $\mathcal{C}$ can be learned using q statistical queries with tolerance τ, then there is a PAC algorithm that learns $\mathcal{C}$ using $m = O(\frac{q\log|\mathcal{C}|}{\tau^2}(\log(q) + \log\log(|\mathcal{C}|))$ samples and $b = O(\log|\mathcal{C}| \cdot \log(\frac{q}{\tau}))$ bits. Theorem 3 gives an SQ learning algorithm $q = \mathrm{poly}(d/\epsilon)$ and $\tau \geq \mathrm{poly}(\epsilon/d)$, which gives a bounded memory learning algorithm with $m = \mathrm{poly}(d/\epsilon) \cdot \log|\mathcal{C}| \cdot \log\log|\mathcal{C}|$ samples and $b = O(\log|\mathcal{C}| \cdot \log(d/\epsilon))$ bits.

□

4 From bounded memory learning to bounded SQ dimension

In this section we prove our lower bounds: Theorem 2 and Theorem 4. A schematic illustration of the proof is given in Fig. 2.

Overview. We use the rejection sampling technique to transform a given strong learner with respect to distribution P into a weak learner with respect to any close distribution Q. This can be established

both in the PAC learning framework and the SQ framework. By virtue of Proposition 2, this implies Theorem 4. To prove Theorem 2, we would like to use a recent result by [17] that establishes an upper bound on $\mathrm{SQ}_Q(\mathcal{C})$ given memory-efficient learner. Unfortunately, the derivation in [17] requires the learner to return the *exact* target concept. Our weak learner does not necessarily satisfy this requirement. In fact, it is even not necessarily proper, i.e., it might return a hypothesis $h \notin \mathcal{C}$. To get around this obstacle, we first show how to transform any improper weak learning rule into a proper learning rule. Then, we focus on the hypotheses $\mathcal{H} \subseteq \mathcal{C}$ that constituents that SQ dimension, i.e, $SQ_Q(\mathcal{H}) = SQ_Q(\mathcal{C})$. We ensure that the exact target concept c is returned, as large $SQ_Q(\mathcal{H})$ implies that all hypotheses in $\mathcal{H}$ are far a part.

From strong learning to weak learning of close distributions. The next claim shows that if a class is strongly learnable under distribution P, then it is weakly learnable under *any* close distribution Q. The idea is to utilize the closeness assumption in order to perform rejection sampling from Q to simulate sampling from P.

Lemma 2. *Let P be a distribution over $\mathcal{X}$. Assume that the concept class $\mathcal{C}$ can be learned with accuracy $1 - 0.1\epsilon$, m samples, and b bits under distribution P. Then, any probability distribution Q that is $(1/\epsilon)$-close to P can be learned with accuracy 0.9, $O(m/\epsilon^2)$ samples, and b bits.*

Rejection sampling algorithm in the SQ model. Analogously to Lemma 2, we can show that also under the SQ framework, strong learning implies weak learning of close distributions. The proof uses the same rejection sampling technique as in Lemma 2.

Lemma 3. *Let P be a distribution over $\mathcal{X}$. Assume the concept class $\mathcal{C}$ can be learned with accuracy $1 - 0.1\epsilon$, q queries and tolerance τ under distribution P. Then, any probability distribution Q that is $(1/\epsilon)$-close to P can be SQ-learned with accuracy 0.9 using $O(q/\epsilon\tau)$ queries with tolerance $\epsilon\tau/2$.*

From weak learning to low SQ-dimension. The next few claims establish the fact that if a class $\mathcal{C}$ is learnable with bounded memory under distribution Q, the statistical dimension $SQ_Q(\mathcal{C})$ is low.

Proposition 5 (Corollary 8 in [17])**.** *Let $\mathcal{H} = \{h_1, \ldots, h_d\}$ a class and Q a distribution with $SQ_Q(\mathcal{H}) = d$. Any learning algorithm that uses m samples, b bits and returns the exact correct hypothesis with probability at least $\Omega(1/m)$ must use at least $m = d^{\Omega(1)}$ samples or $\Omega(\log^2 d)$ bits.*[3]

The algorithm described in the previous section will not return the exact hypothesis, and more generally will not even be a proper learner (i.e., it will not necessarily return a hypothesis from the class). Fortunately, we can transform any improper learner into a proper learner without significantly increasing the neither the sample nor the space complexity.

Lemma 4. *Fix a class $\mathcal{C}$. Let $\mathcal{A}$ be an improper learning algorithm for $\mathcal{C}$ that uses b bits, m samples, and accuracy $1 - \epsilon$. Then there is an $(1 - 3\epsilon)$-accurate* proper *learning algorithm that uses $O(m)$ samples and $b + O(\log(|\mathcal{C}|/\epsilon))$ bits.*

Lemma 5. *Fix a class $\mathcal{C}$ and a distribution Q. If $\mathcal{C}$ is learnable with accuracy 0.9 under Q using m samples and b bits, then*

$$SQ_Q(\mathcal{C}) \leq \max(m^{O(1)}, 2^{O(\sqrt{b})}).$$

Putting it all together. We now complete the proofs of Theorem 2 and Theorem 4.

Proof of Theorem 2. Assume the concept class $\mathcal{C}$ can be learned with accuracy $1 - 0.1\epsilon$, m samples, and b bits under distribution P. Lemma 2 states that any distribution Q that is $(1/\epsilon)$-close to P can be learned with accuracy 0.9, $O(m/\epsilon^2)$ samples, and b bits. Lemma 5 completes the claim. □

Proof of Theorem 4. Assume that the concept class $\mathcal{C}$ can be learned with accuracy $1-0.1\epsilon$, q queries and tolerance τ under distribution P. Lemma 3 states that any distribution $Q \in \mathcal{P}_{1/\epsilon}(P)$ can be SQ-learned with accuracy 0.9, $O(m/\epsilon\tau)$ queries, and tolerance $\epsilon\tau/2$. Proposition 2 completes the claim. □

[3] In [17] they consider the case where Q is the uniform distribution. By creating a few copies of the examples in $\mathcal{X}$ we can transform a general *known* distribution to be as close as to uniform as needed. Note that the size of the domain $\mathcal{X}$ is not a relevant parameter here.

Broader Impact

Algorithms with bounded memory are extensively studied ([1],[42],[28]). But bounded memory *learning* algorithms were only recently been investigated. In machine learning we have a good understanding of PAC learning using the VC dimension; weak learning with statistical queries using the SQ dimension; and online learning using the Littlestone dimension. An understanding of bounded-memory learning is missing. There are many works showing lower bounds, but none that shows both upper and lower bounds.

We are the first to (1) give a characterization of bounded-memory learning in some regime, (2) in this regime we show equivalence to a different and known framework, statistical queries.

Our impact is two-fold: for the general ML community we give an understanding of the capabilities and limitations of bounded-memory learning and we show its equivalence to a known framework. Second, for the theory researchers, we leave many open problems:

1. Proving a characterization for the entire regime.
2. Utilizing the equivalence between statistical queries and bounded memory to gain a better understanding of these two frameworks.
3. Our work focused on the case that $|C|, |X|$ are polynomially related. We leave for future research to investigate the regimes of $|C| = |X|^{o(1)}$ and $|X| = |C|^{o(1)}$.

Acknowledgments and Disclosure of Funding

This work is supported by NSF award 1909634

References

[1] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[2] Javed A Aslam and Scott E Decatur. General bounds on statistical query learning and PAC learning with noise via hypothesis boosting. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 282–291. IEEE, 1993.

[3] José L Balcázar, Jorge Castro, David Guijarro, Johannes Köbler, and Wolfgang Lindner. A general dimension for query learning. *Journal of Computer and System Sciences*, 73(6):924–940, 2007.

[4] Paul Beame, Shayan Oveis Gharan, and Xin Yang. Time-space tradeoffs for learning finite functions from random evaluations, with applications to polynomials. In *Conference On Learning Theory*, pages 843–856, 2018.

[5] Shai Ben-David, Tyler Lu, and Dávid Pál. Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning. In *COLT*, pages 33–44, 2008.

[6] Gyora M Benedek and Alon Itai. Learnability with respect to fixed distributions. *Theoretical Computer Science*, 86(2):377–389, 1991.

[7] Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using fourier analysis. In *STOC*, volume 94, pages 253–262, 1994.

[8] Avrim Blum, John Hopcroft, and Ravindran Kannan. Foundations of data science. *Vorabversion eines Lehrbuchs*, 2016.

[9] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

[10] Nader H Bshouty and Dmitry Gavinsky. On boosting with polynomially bounded distributions. *Journal of Machine Learning Research*, 3(Nov):483–506, 2002.

[11] Yuval Dagan, Gil Kur, and Ohad Shamir. Space lower bounds for linear prediction in the streaming model. In *Conference on Learning Theory*, pages 929–954, 2019.

[12] Yuval Dagan and Ohad Shamir. Detecting correlations with little memory and communication. In *Conference On Learning Theory*, pages 1145–1198, 2018.

[13] Vitaly Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer and System Sciences*, 78(5):1444–1459, 2012.

[14] Yoav Freund. An improved boosting algorithm and its implications on learning complexity. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 391–398. ACM, 1992.

[15] Yoav Freund. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.

[16] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[17] Sumegha Garg, Ran Raz, and Avishay Tal. Extractor-based time-space lower bounds for learning. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 990–1002. ACM, 2018.

[18] Sumegha Garg, Ran Raz, and Avishay Tal. Time-space lower bounds for two-pass learning. In *34th Computational Complexity Conference (CCC 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[19] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 538–545. IEEE, 1995.

[20] Jeffrey C Jackson. The harmonic sieve: A novel application of fourier analysis to machine learning theory and practice. Technical report, Carnegie Mellon University Pittsburgh School Of Computer Science, 1995.

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

[22] Adam R Klivans and Rocco A Servedio. Boosting and hard-core sets. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 624–633. IEEE, 1999.

[23] Gillat Kol, Ran Raz, and Avishay Tal. Time-space hardness of learning sparse parities. In *Proc. 49th ACM Symp. on Theory of Computing*, 2017.

[24] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

[25] Dana Moshkovitz and Michal Moshkovitz. Mixing implies lower bounds for space bounded learning. In *Conference on Learning Theory*, pages 1516–1566, 2017.

[26] 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)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[27] Michal Moshkovitz and Naftali Tishby. A general memory-bounded learning algorithm. *arXiv preprint arXiv:1712.03524*, 2017.

[28] Shanmugavelayutham Muthukrishnan et al. Data streams: Algorithms and applications. *Foundations and Trends® in Theoretical Computer Science*, 1(2):117–236, 2005.

[29] Ran Raz. Fast learning requires good memory: A time-space lower bound for parity learning. In *Proc. 57th IEEE Symp. on Foundations of Computer Science*, 2016.

[30] Ran Raz. A time-space lower bound for a large class of learning problems. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 732–742. IEEE, 2017.

[31] Sivan Sabato, Nathan Srebro, and Naftali Tishby. Distribution-dependent sample complexity of large margin learning. *The Journal of Machine Learning Research*, 14(1):2119–2149, 2013.

[32] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[33] Robert E Schapire. The design and analysis of efficient learning algorithms. Technical report, Massachusetts Inst Of Tech Cambridge Lab For Computer Science, 1991.

[34] O. Shamir. Fundamental limits of online and distributed algorithms for statistical learning and estimation. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 163–171, 2014.

[35] Vatsal Sharan, Aaron Sidford, and Gregory Valiant. Memory-sample tradeoffs for linear regression with small error. *arXiv preprint arXiv:1904.08544*, 2019.

[36] Hans Ulrich Simon. A characterization of strong learnability in the statistical query model. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 393–404. Springer, 2007.

[37] Jacob Steinhardt, Gregory Valiant, and Stefan Wager. Memory, communication, and statistical queries. In *Conference on Learning Theory*, pages 1490–1516, 2016.

[38] Balázs Szörényi. Characterizing statistical query learning: simplified notions and proofs. In *International Conference on Algorithmic Learning Theory*, pages 186–200. Springer, 2009.

[39] Leslie G Valiant. A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 436–445. ACM, 1984.

[40] Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer, 2015.

[41] Nicolas Vayatis and Robert Azencott. Distribution-dependent vapnik-chervonenkis bounds. In *European Conference on Computational Learning Theory*, pages 230–240. Springer, 1999.

[42] Avi Wigderson. Mathematics and computation. *IAS, Draft (March 2018)*, 2018.

[43] Ke Yang. On learning correlated boolean functions using statistical queries. In *International Conference on Algorithmic Learning Theory*, pages 59–76. Springer, 2001.

[44] Ke Yang. New lower bounds for statistical query learning. *Journal of Computer and System Sciences*, 70(4):485–509, 2005.

A Background

PAC model. In PAC learning [39] we consider the task of binary classification over an *instance space* $\mathcal{X}$. Denote by $\mathcal{C} \subseteq \{-1,1\}^{\mathcal{X}}$ a concept class of functions mapping instances to binary labels, and let $c \in \mathcal{C}$ be the *target* (a.k.a. true) concept. Also, let P be the underlying probability distribution over $\mathcal{X}$. We assume that P is known to the learner whereas the target concept c is not known.

The input to the learning algorithm $\mathcal{A}$ consists of a labeled sample $S = ((x_1, c(x_1)), \ldots, (x_m, c(x_m)))$ such that $S_X := (x_1, \ldots, x_m) \sim P^m$. Its output has the form of a hypothesis $h \in \{-1,1\}^{\mathcal{X}}$. We measure the success of the algorithm according to its expected error $L_{P,c}(h) = \Pr_{x\sim P}(h(x) \neq c(x))$. We say that h is ϵ-accurate if $L_{P,c}(h) \leq \epsilon$. The *sample complexity* of $\mathcal{A}$ under the distribution P, denoted $m(\epsilon) : (0,1) \to \mathbb{N}$, is a function mapping a desired accuracy ϵ to the minimal positive integer $m(\epsilon)$ such that for any target concept $c \in \mathcal{C}$ and any $m \geq m(\epsilon)$, with probability at least $2/3$ over the drawn of an i.i.d. sample $S = ((x_1, c(x_1)), \ldots, (x_m, c(x_m)))$, the output $\mathcal{A}(S)$ is ϵ-accurate.[4]

The statistical query framework. The statistical query (SQ) framework has been introduced by [21] to handle random noise in the PAC setting. In this model, instead of having access to an i.i.d. sequence of labeled instances, the learner has access to a *statistical query oracle* (a.k.a. *correlation oracle*). Each call to the oracle has the form of a pair (h, τ), where $h \in \{-1,1\}^{\mathcal{X}}$ is a hypothesis and $\tau > 0$ is called a *tolerance* parameter. The oracle has to answer such a query with a scalar ν satisfying[5]

$$|\langle h, c\rangle_P - \nu| \leq \tau \quad \text{where} \quad \langle h, c\rangle_P := \mathbb{E}_{x\sim P}[h(x)c(x)].$$

As was shown in [21], any approximately accurate algorithm in the SQ model can be efficiently transformed into an approximately accurate PAC algorithm, i.e. an algorithm that has access to i.i.d. labeled examples. The resulted PAC is also robust to noise. We refer to [38] for additional background.

Analogously to the definition of sample complexity, the *query complexity* of a learning algorithm in the SQ model, denoted $q_\tau(\epsilon)$, is the minimal number of queries with tolerance parameter τ required for achieving ϵ-accurate prediction (for any target concept $c \in \mathcal{C}$).

SQ dimension. The SQ-dimension defined below is useful for characterizing weak learnability in the statistical query framework, as was proved in [7] (see Proposition 1 and Proposition 2).

Definition 3 (Statistical query dimension)**.** *Fix a probability distribution P over $\mathcal{X}$. The SQ-dimension of the class $\mathcal{C}$ with respect to the distribution P, denoted $\mathrm{SQ}_P(\mathcal{C})$, is the maximal integer d such that there exist $h_1, \ldots, h_d \in \mathcal{C}$ satisfying $|\langle h_i, h_j\rangle_P| \leq 1/d$ for all $i \neq j \in [d]$.*

Additional notation We denote the density and the cumulative binomial distribution by $\mathrm{Binom}(m, r, p)$ and $\mathrm{Binom}(m, \leq r, p)$, which respectively refer to the probability of observing exactly (at most) r heads in m independent and identical trials where the probability of "head" in each single trial is p.[6]

[4] Given a confidence parameter $\delta > 2/3$, standard amplification techniques can be used to ensure that the probability error is at most δ, while increasing the sample complexity by at most a $\log(1/\delta)$ multiplicative factor.

[5] According to the original framework of Kearns, (seemingly) more general queries are allowed. Namely, each query is a pair (χ, τ) where $\chi : \mathcal{X} \times \{-1,1\} \to \{-1,1\}$. The oracle has to answer the query with a scalar ν satisfying

$$|\mathbb{E}_{x\sim P}[\chi(x, c(x))] - \nu| \leq \tau\ .$$

Note that $\chi(x, c(x))$ can be written as a polynomial in x and $c(x)$, and since $c(x)$ is either 1 or -1, this polynomial is linear in $c(x)$. In other words, $\chi(x, c(x)) = g_1(x)c(x) + g_2(x)$. given that the distribution P is known, $\mathbb{E}_{x\sim P}[g_2(x)]$ can be calculated. Thus, one can simulate the seemingly more general query χ using the correlation query applied to g_1.

[6] If $r > m$ or $r < 0$ then both terms are equal to zero.

B Omitted Proofs

B.1 From bounded SQ dimension to bounded memory learning

Reviewing Boost-By-Majority (BBM). Let $\mathcal{W}$ be a γ-weak learner with respect to the distribution P with sample complexity m_0. Similarly to most boosting algorithms, BBM operates by iteratively re-weighting and feeding a given γ-weak learner with T i.i.d. samples $S_1, \ldots, S_T$ of size m_0. The outputs $h_1, \ldots, h_T$ of the weak learner are then aggregated into a majority vote classifier:

$$h(x) = \texttt{Majority}(h_1(x), \ldots, h_T(x)) := \begin{cases} 1 & \sum_t h_t(x) > 0 \\ -1 & \text{otherwise} \end{cases}.$$

To make the algorithm memory-efficient [15] suggests to implement the re-weighting using *rejection sampling*. Let $h_1, \ldots, h_t$ be the weak classifiers collected during the first t rounds. At the beginning of round $t+1$, the algorithm draws an example $x \sim P$ and keeps it with probability

$$w_{t+1}(x) = \text{Binom}\left(T-t, \left\lfloor \frac{T-t-r(x)}{2} \right\rfloor, 1/2+\gamma\right) \qquad \text{where } r(x) := \sum_{i=1}^{t} h_i(x). \tag{1}$$

Therefore, the induced probability distribution on time t is

$$P_{t+1}(x) = w_{t+1}(x)P(x)/Z \tag{2}$$

where Z is a normalization factor. It repeats this step until either collecting m_0 samples or rejecting $\Theta(\epsilon^{-3}\log T)$ consecutive examples. In the former scenario it feeds the weak learner with the resulted sample, whereas in the latter scenario it aborts the boosting process and returns the hypothesis $h = \texttt{Majority}(h_1(x), \ldots, h_t(x))$.[7]

Proposition 6. *[15] Let $\epsilon > 0$. With probability at least $2/3$, the following hold:*

1. *BBM reaches an ϵ-accurate hypothesis after at most $T = O(\gamma^{-2}\log(1/\epsilon))$ rounds.*
2. *There exists a global constant $C > 0$ such that for every round t, the probability distribution P_t satisfies $P_t(x) \le (C/\epsilon^3) \cdot P(x)$ for all x.*

Proof of Lemma 1. Let $\delta = 1/\mu$. Consider the mixed distribution $\tilde{P}_t = \delta P + (1-\delta)P_t$. Proposition 6 implies that for all x, $P_t(x) \le \mu P(x)$. It follows that

$$(\forall x) \qquad \tilde{P}_t(x) \le \delta P(x) + (1-\delta)\mu P(x) \le \mu P(x).$$

Also, clearly we have that

$$(\forall x) \qquad \tilde{P}_t(x) \ge \delta P(x) = \mu^{-1} P(x).$$

Hence, $\tilde{P}_t \in \mathcal{P}_\mu(P)$, and by our assumption we have $\mathrm{SQ}_{\tilde{P}_t}(\mathcal{C}) \le d$.

Assume by contradiction that there exist $m \ge 4d$ hypotheses $h_1, \ldots, h_m \in \mathcal{C}$ such that

$$|\langle h_i, h_j \rangle_{P_t}| \le 1/m \qquad (\forall i \ne j \in [m]).$$

Therefore, for all $i \ne j \in [m]$,

$$\left|\langle h_i, h_j \rangle_{\tilde{P}_t}\right| = |\delta \langle h_i, h_j \rangle_P + (1-\delta)\langle h_i, h_j \rangle_{P_t}| \le \delta + (1-\delta)\frac{1}{m} \le \frac{1}{4d} + \frac{1}{4d} = \frac{1}{2d}.$$

In particular, it follows that $|\langle h_i, h_j \rangle_{\tilde{P}_t}| \le \frac{1}{2d}$ for all $i \ne j \in [2d]$. This contradicts the fact that $\mathrm{SQ}_{\tilde{P}_t}(\mathcal{C}) \le d$. □

[7] In [15], the algorithm does not actually abort but proceeds by drawing random hypotheses for $T-t$ rounds. It was shown in [20], Lemma 5.2, that (with the above rejection criteria) the algorithm can actually abort and return a majority vote.

B.2 From bounded memory learning to bounded SQ dimension

Proof of Lemma 2. Fix a distribution P, a class $\mathcal{C}$ and assume that there is an algorithm $\mathcal{A}$ that learns $\mathcal{C}$ under P with accuracy $1 - 0.1\epsilon$, m samples, and b bits. We want to show that for any $(1/\epsilon)$-close distribution $Q \in \mathcal{P}_{1/\epsilon}(P)$ there is an algorithm that learns the class $\mathcal{C}$ under distribution Q with accuracy 0.9, $O(m/\epsilon^2)$ samples, and b bits.

At a high level, our analysis involves two steps. First, given a close distribution Q we apply the rejection sampling technique to simulate sampling from the original distribution P. This enables us to run the algorithm $\mathcal{A}$. Then we translate the accuracy guarantee of $\mathcal{A}$ with respect to P into a an accuracy guarantee with respect to Q.

Rejection sampling. In Algorithm 1 we detail the rejection sampling step mentioned above.

Algorithm 1 Learning from examples distributed by Q

1: Get a labeled example x from Q.
2: Accept x with probability $\frac{P(x)}{Q(x)}\epsilon$.
3: Call algorithm $\mathcal{A}$ with the accepted examples.

We first note that the rejection sampling is well defined. Namely, by the closeness assumption, $\frac{P(x)}{Q(x)}\epsilon \in [0, 1]$. The distribution induced by the rejection sampling is proportional to P since

$$Q(x) \cdot \frac{P(x)}{Q(x)}\epsilon = P(x)\epsilon.$$

Strong learning with respect to $P \Rightarrow$ weak learning with respect to Q. By our assumption on $\mathcal{A}$, with probability at least $2/3$, it outputs a hypothesis h with accuracy at least $1 - 0.1\epsilon$. We next prove that h forms a weak classifier with respect to Q. Denoting the target hypothesis by $c \in \mathcal{C}$, we have that

$$L_{Q,c}(h) = \sum_{x:h(x)\neq c(x)} Q(x) \leq \sum_{x:h(x)\neq c(x)} \frac{1}{\epsilon} \cdot P(x) = \frac{1}{\epsilon} \cdot L_{P,c}(h) \leq \frac{1}{\epsilon} \cdot 0.1\epsilon = 0.1 \ .$$

Thus, the accuracy is at least 0.9.

So far we proved that the we indeed designed a learning algorithm for Q. Let's analyze the parameters of the algorithm. The rejection sampling technique does not require additional bits, thus number of bits is the same as number of bits used in $\mathcal{A}$. We next bound the number of samples needed.

We first note that the probability to accept an example x is $\frac{P(x)}{Q(x)}\epsilon \geq \epsilon^2$, as Q is $(1/\epsilon)$-close to P. From Hoeffding's inequality, we know that if we get at least $2m/\epsilon^2$ samples, then the probability that the algorithm does not accept at least m samples is smaller than e^{-m}. Thus, with probability at least $1 - me^{-m}$, the number of samples used by the new algorithm is $O(m/\epsilon^2)$.

The confidence of the algorithm is at least $2/3 - e^{-m} \cdot m \geq 7/12$ for large enough m. Standard amplification techniques can be used to ensure that the probability error is at most $2/3$, while increasing the sample complexity by at most a constant multiplicative factor. □

Proof of Lemma 3. Fix a distribution P, a class $\mathcal{C}$ and assume that there is an algorithm $\mathcal{A}$ that learns $\mathcal{C}$ under P with accuracy $1 - 0.1\epsilon$, m queries, and tolerance τ. Denote the correct hypothesis by $c \in \mathcal{C}$. We want to show that for any $(1/\epsilon)$-close distribution $Q \in \mathcal{P}_{1/\epsilon}(P)$ there is an algorithm that weakly learns the class C under distribution Q in the SQ framework.

Fix a query ψ that is used by $\mathcal{A}$. Ideally, we would like to replace it with a query ψ' of the form

$$\psi'(x) = \begin{cases} \frac{P(x)}{Q(x)}\psi(x) & \text{if } Q(x) \neq 0 \\ 0 & otherwise \end{cases} \ ,$$

since querying ψ under P is the same as querying ψ' under Q, as $\mathbb{E}_Q[\psi'(x)c(x)] = \mathbb{E}_P[\psi(x)c(x)]$. The problem is that the range of ψ' is not $\{-1, 1\}$. To fix it, we will replace ψ with several queries

$\psi_1, \ldots, \psi_n$ that their range is $\{-1, 1\}$ and their average, $\frac{1}{n}\sum_{i=1}^{n}\psi_i$, approximately returns the correct query, i.e., $\psi' \approx \frac{1}{n}\sum_{i=1}^{n}\psi_i$.

For every $x \in \mathcal{X}$ we would like to use Lemma 6 below in order to define $\psi_i(x)$. The first step will be to make sure that $\psi'(x)$ is in $[-1, 1]$. To achieve that we focus on $\epsilon\psi'(x)$, because it is equal to $\epsilon\frac{P(x)}{Q(x)}\psi(x)$ and

$$0 < \epsilon \cdot \frac{P(x)}{Q(x)} \leq \epsilon \cdot \frac{1}{\epsilon} = 1.$$

Using Lemma 6, there are $n = O(1/\epsilon\tau)$ queries ψ_i such that for every $x \in \mathcal{X}$ it holds that

$$\left|\frac{1}{n}\sum_{i=1}^{n}\psi_i(x) - \epsilon\psi'(x)\right| \leq \frac{\epsilon\tau}{2}.$$

From this we can deduce that

$$\left|\frac{1}{\epsilon}\cdot\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_Q[\psi_i(x)c(x)] - \mathbb{E}_P[\psi(x)c(x)]\right| \leq \frac{\tau}{2}.$$

To summarize, the new learning algorithm $\mathcal{A}'$ that learns under distribution Q will simulate algorithm $\mathcal{A}$ and whenever a query ψ will be needed, it will take $O(1/\epsilon\tau)$ queries created by Lemma 6 and return their average times $1/\epsilon$. Thus, $\mathcal{A}$ uses $O(m/\epsilon\tau)$ queries and its tolerance is $\epsilon\tau/2$.

□

Proof of Lemma 4. Fix a class $\mathcal{C}$ and an improper learning algorithm $\mathcal{A}$ for $\mathcal{C}$. Denote the number of bits it uses by b, the number of samples by m, and the accuracy by $1-\epsilon$. Define the algorithm $\mathcal{A}'$ as follows:

1. Run algorithm $\mathcal{A}$ that outputs hypothesis h as its answer.
2. Go over all hypothesis in $\mathcal{C}$ and return one that agrees with h on $1-2\epsilon$ of the examples by testing consistency on $O(\log|\mathcal{C}|/\epsilon^2)$ random examples.

Note that the second step does not use new samples and requires only $\log|\mathcal{C}| + O(\log(\log|\mathcal{C}|/\epsilon)) = O(\log(|\mathcal{C}|/\epsilon))$ additional bits. The algorithm $\mathcal{A}'$ functions correctly, because by the definition of the algorithm $\mathcal{A}$ there must be hypothesis in $\mathcal{C}$ that agrees on $(1-\epsilon)$ of the examples. By Hoeffding's inequality, the probability that there is a hypothesis that deviates by more than ϵ in approximating its loss is small and standard amplification techniques can be used to ensure that the probability error is at most $2/3$, while increasing the sample complexity by at most a constant multiplicative factor. The accuracy of $\mathcal{A}'$ is at least $1-3\epsilon$. □

Proof of Lemma 5. Fix a class $\mathcal{C}$ and a distribution Q. Assume $\mathcal{C}$ is learnable under Q with m samples, b bits, and accuracy 0.9. Assume also that $SQ_Q(\mathcal{C}) = d$. Thus, there are d hypotheses $\mathcal{H} = \{h_1, \ldots, h_d\}$ such that $|\langle h_i, h_j\rangle_Q| \leq 1/d$. Since $\mathcal{H} \subseteq \mathcal{C}$ and by our assumption on the learnability of $\mathcal{C}$, we get that $\mathcal{H}$ is learnable under Q with m samples, b bits, and accuracy 0.9. From Lemma 4, we get that $\mathcal{H}$ is *properly* learnable under Q with $O(m)$ samples, $b + O(\log|\mathcal{H}|)$ bits, and accuracy 0.7.

We can deduce that there is a learning algorithm for $\mathcal{H}$ that returns the exact hypothesis, as the hypotheses in $\mathcal{H}$ are far apart from each other. Specifically, we know that between any two hypotheses $i \neq j$ there is at least $\frac{1}{2} - \frac{1}{2d}$ disagreement. If $\frac{1}{2} - \frac{1}{2d} > 0.3$, then learning exactly is equivalent to properly learning up to accuracy 0.7. The equation $\frac{1}{2} - \frac{1}{2d} > 0.3$ is equivalent to $d = \Omega(1)$.

Since the hypotheses in $\mathcal{H}$ are far apart from each other, the number of bits $\mathcal{A}$ uses is lower bounded by $b \geq \log|\mathcal{H}|$, as the hypothesis in $\mathcal{H}$ returned by the algorithm must be computed from its internal state. Thus the memory requirement of the proper learning algorithm is $O(b)$ bits.

Now we can apply Proposition 5, as for large enough constant M, for $m \geq M$, the probability to succeed, $2/3$, is $\Omega(1/m)$. We get that $m = d^{\Omega(1)}$ or $b = \Omega(\log^2 d)$. Equivalently, $d = m^{O(1)}$ or $d = 2^{O(\sqrt{b})}$. In other words, $SQ_Q(\mathcal{C}) \leq \max(m^{O(1)}, 2^{O(\sqrt{b})})$. □

Lemma 6. *For any $\gamma \in [-1, 1]$ and $\tau \in (0, 1]$, there are $n = O(1/\tau)$ numbers $y_1, \ldots, y_n \in \{-1, 1\}$ such that $|\frac{1}{n}\sum_i y_i - \gamma| \leq \tau$.*

Proof. Take n such that $1/n < \tau$. Let $k \in \{0, 1, \ldots, n\}$ be such that $(n - 2k)/n$ is $1/n$ close to γ. Take $y_1 = \ldots = y_k = -1$ and $y_{k+1} = \ldots = y_n = 1$. □