SIGACT News Complexity Theory Column 95

Lane A. Hemaspaandra
Dept. of Computer Science
University of Rochester
Rochester, NY 14627, USA

Introduction to Complexity Theory Column 95

This block of text is the amount of space you need to please leave for Lane's introduction. This block of text is the amount of space you need to please leave for Lane's introduction. This block of text is the amount of space you need to please leave for Lane's introduction. This block of text is the amount of space you need to please leave for Lane's introduction. This block of text is the amount of space you need to please leave for Lane's introduction.

Guest Column: A Survey of Quantum Learning Theory[1]

Srinivasan Arunachalam[2] *and Ronald de Wolf*[3]

[1] © Srinivasan Arunachalam and Ronald de Wolf, 2017

[2] QuSoft, CWI, Amsterdam, the Netherlands. Supported by ERC Consolidator Grant QPROGRESS 615307.

[3] QuSoft, CWI and University of Amsterdam, the Netherlands. Partially supported by ERC Consolidator Grant QPROGRESS 615307.

Abstract. This paper surveys quantum learning theory: the theoretical aspects of machine learning using quantum computers. We describe the main results known for three models of learning: exact learning from membership queries, and Probably Approximately Correct (PAC) and agnostic learning from classical or quantum examples.

1 Introduction

Machine learning entered theoretical computer science in the 1980s with the work of Leslie Valiant [Val84], who introduced the model of "Probably Approximately Correct" (PAC) learning, building on earlier work of Vapnik and others in statistics, but adding computational complexity aspects. This provided a mathematically rigorous definition of what it means to (efficiently) learn a target concept from given examples. In the three decades since, much work has been done in computational learning theory: some efficient learning results, many hardness results, and many more models of learning. We refer to [KV94b, AB09, SB14] for general introductions to this area. In recent years practical machine learning has gained an enormous boost from the success of deep learning in important big-data tasks like image recognition, natural language processing, and many other areas; this is theoretically still not very well understood, but it often works amazingly well.

Quantum computing started in the 1980s as well, with suggestions for analog quantum computers by Manin [Man80, Man99], Feynman [Fey82, Fey85], and Benioff [Ben82], and reached more digital ground with Deutsch's definition of a universal quantum Turing machine [Deu85]. The field gained momentum with Shor's efficient quantum algorithms [Sho97] for factoring integers and computing discrete logarithms (which between them break much of today's public-key cryptography), and has since blossomed into a major area at the crossroads of physics, mathematics, and computer science.

Given the successes of both machine learning and quantum computing, combining these two strands of research is an obvious direction. Indeed, soon after Shor's algorithm, Bshouty and Jackson [BJ99] introduced a version of learning from *quantum* examples, which are quantum superpositions rather than random samples. They showed that Disjunctive Normal Form (DNF) can be learned efficiently from quantum examples under the uniform distribution; efficiently learning DNF from uniform *classical* examples (without membership queries) was and is an important open problem in classical learning theory. Servedio and others [AS05, AS09, SG04] studied upper and lower bounds on the number of quantum membership queries or quantum examples needed for learning, and more recently the authors of the present survey obtained optimal bounds on quantum sample complexity [AW16].

Focusing on specific learning problems where quantum algorithms may help, Aïmeur et al. [ABG06, ABG13] showed quantum speed-up in learning contexts such as clustering via minimum spanning tree, divisive clustering, and k-medians, using variants of Grover's search algorithm [Gro96]. In the last few years there has been a flurry of interesting results applying various quantum algorithms (Grover's algorithm, but also phase estimation, amplitude amplification [BHMT02], and the HHL algorithm for solving well-behaved systems of linear equations [HHL09]) to specific machine learning problems. Examples include Principal Component Analysis [LMR13b], support vector machines [RML13], k-means clustering [LMR13a], quantum recommendation systems [KP17], and work related to neural networks [WKS16b, WKS16a]. Some of this work—like most of application-oriented machine learning in general—is heuristic in nature rather than mathematically rigorous. Some of these new approaches are suggestive of exponential speed-ups over classical machine learning, though one has to be careful about the underlying assumptions needed to make efficient quantum machine learning possible: in some cases these also make efficient *classical* machine learning possible. Aaronson [Aar15] gives a brief but clear description of the issues. These developments have been well-served by a number of recent survey papers [SSP15, AAD$^+$15, BWP$^+$16] and even a book [Wit14].

In contrast, in this survey we focus on the theoretical side of quantum machine learning: quantum learning theory.[4] We will describe (and sketch proofs of) the main results that have been obtained in three main learning models. These will be described in much more detail in the next sections, but below we give a brief preview.

Exact learning. In this setting the goal is to learn a target concept from the ability to interact with it. For concreteness, we focus on learning target concepts that are Boolean functions: the target is some unknown $c : \{0,1\}^n \to \{0,1\}$ coming from a known concept class $\mathcal{C}$ of functions,[5] and our goal is to identify c exactly, with high probability, using *membership queries* (which allow the learner to learn $c(x)$ for x of his choice). If the measure of complexity is just the number of queries, the main results are that quantum exact learners can be polynomially more efficient than classical, but not more. If the measure of complexity is *time*, then under reasonable complexity-theoretic assumptions some concept classes can be learned much faster from quantum membership queries (i.e., where the learner can query c on a superposition of x's) than is possible classically.

PAC learning. In this setting one also wants to learn an unknown $c : \{0,1\}^n \to \{0,1\}$ from a known concept class $\mathcal{C}$, but in a more passive way than with membership queries: the learner receives several *labeled examples* $(x, c(x))$, where x is distributed according to some unknown probability distribution D over $\{0,1\}^n$. The learner gets multiple i.i.d. labeled examples. From this limited "view" on c, the learner wants to generalize, producing a *hypothesis* h that probably agrees with c on "most" x, *measured according to the same* D. This is the classical Probably Approximately Correct (PAC) model. In the quantum PAC model [BJ99], an example is not a random sample but a *superposition* $\sum_{x\in\{0,1\}^n} \sqrt{D(x)}|x, c(x)\rangle$. Such quantum examples can be useful for some learning tasks with a fixed distribution D (e.g., uniform D) but it turns out that in the usual distribution-independent PAC model, quantum and classical sample complexity are equal up to constant factors, for every concept class $\mathcal{C}$. When the measure of complexity is *time*, under reasonable complexity-theoretic assumptions, some concept classes can be PAC learned much faster by quantum learners (even from classical examples) than is possible classically.

Agnostic learning. In this setting one wants to approximate a distribution on $\{0,1\}^{n+1}$ by finding a good hypothesis h to predict the last bit from the first n bits. A "good" hypothesis is one that is not much worse than the best predictor available in a given class $\mathcal{C}$ of available hypotheses. The agnostic model has more freedom than the PAC model and allows to model more realistic

[4] The only other paper we are aware of to survey quantum learning theory is an unpublished manuscript by Robin Kothari from 2012 [Kot12] which is much shorter but partially overlaps with ours; we only saw this after finishing a first version of our survey.

[5] Considering concept classes over $\{0,1\}^n$ has the advantages that the n-bit x in a labeled example $(x, c(x))$ may be viewed as a "feature vector". This fits naturally when one is learning a type of objects characterized by patterns involving n features that each can be present or absent in an object, or when learning a class of n-bit Boolean functions such as small decision trees, circuits, or DNFs. However, we can (and sometimes do) also consider concepts $c : [N] \to \{0,1\}$.

situations, for example when the data is noisy or when no "perfect" target concept exists. Like in the PAC model, it turns out quantum sample complexity is not significantly smaller than classical sample complexity in the agnostic model.

Organization. The survey is organized as follows. In Sections 2 and 3 we first introduce the basic notions of quantum and learning theory, respectively. In Section 4 we describe the main results obtained for information-theoretic measures of learning complexity, namely query complexity of exact learning and sample complexities of PAC and agnostic learning. In Section 5 we survey the main results known about *time* complexity of quantum learners. We conclude in Section 6 with a summary of the results and some open questions for further research.

2 Introduction to quantum information

2.1 Notation

For a general introduction to quantum information and computation we refer to [NC00]. In this survey, we assume familiarity with the following notation. Let $|0\rangle = \binom{1}{0}$ and $|1\rangle = \binom{0}{1}$ be the standard basis states for $\mathbb{C}^2$, the space in which one qubit "lives". Multi-qubit basis states are obtained by taking tensor products of one-qubit basis states; for example, $|0\rangle \otimes |1\rangle \in \mathbb{C}^4$ denotes a basis state of a 2-qubit system where the first qubit is in state $|0\rangle$ and the second qubit is in state $|1\rangle$. For $b \in \{0,1\}^k$, we often shorthand $|b_1\rangle \otimes \cdots \otimes |b_k\rangle$ as $|b_1 \cdots b_k\rangle$. A k-qubit *pure* state $|\phi\rangle$ can be written as $|\phi\rangle = \sum_{i \in \{0,1\}^k} \alpha_i |i\rangle$ where the α_i's are complex numbers (called *amplitudes*) that have to satisfy $\sum_{i \in \{0,1\}^k} |\alpha_i|^2 = 1$. We view $|\phi\rangle$ as a 2^k-dimensional column vector. The row vector that is its complex conjugate is denoted by $\langle\phi|$. An r-dimensional *quantum state* ρ (also called a *density matrix*) is an $r \times r$ positive semi-definite (psd) matrix ρ with trace 1; this can be written (often non-uniquely) as $\rho = \sum_i p_i |\phi_i\rangle\langle\phi_i|$ and hence can be viewed as a probability distribution over pure states $|\phi_i\rangle$.

Non-measuring quantum operations correspond to *unitary* matrices U, which act by left-multiplication on pure states $|\psi\rangle$ (yielding $U|\psi\rangle$), and by conjugation on mixed states ρ (yielding $U\rho U^{-1}$). For example, the 1-qubit Hadamard transform $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ corresponds to the unitary map $H : |a\rangle \to (|0\rangle + (-1)^a |1\rangle)/\sqrt{2}$, for $a \in \{0,1\}$.

To obtain classical information from a quantum state ρ, one can apply a *quantum measurement* to ρ. An m-outcome quantum measurement, also called a *POVM* (positive-operator-valued measure), is described by a set of positive semi-definite matrices $\{M_i\}_{i \in [m]}$ that satisfy $\sum_i M_i = \text{Id}$. When measuring ρ using this POVM, the probability of outcome j is given by $\text{Tr}(M_j \rho)$.

2.2 Query model

In the query model of computation, the goal is to compute a Boolean function $f : \{0,1\}^N \to \{0,1\}$ on some input $x \in \{0,1\}^N$. We are not given x explicitly, instead we are allowed to *query* an oracle that encodes the bits of x, i.e., given $i \in [N]$, the oracle returns x_i. The cost of a query algorithm is the number of queries the algorithm makes to the oracle. We will often assume for simplicity that N is a power of 2, $N = 2^n$, so we can identify indices i with their binary representation $i_1 \dots i_n \in \{0,1\}^n$. Formally, a quantum query corresponds to the following unitary map on $n+1$ qubits:

$$O_x : |i, b\rangle \to |i, b \oplus x_i\rangle,$$

where $i \in \{0, \dots, N-1\}$ and $b \in \{0,1\}$. Given access to an oracle of the above type, we can make a phase query of the form $O_{x,\pm} : |i\rangle \to (-1)^{x_i}|i\rangle$ as follows: start with $|i,1\rangle$ and apply the Hadamard transform to the last qubit to obtain $|i\rangle|-\rangle$ where $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. Apply O_x to $|i\rangle|-\rangle$ to obtain $(-1)^{x_i}|i\rangle|-\rangle$. Finally, apply Hadamard transform to the last qubit to send it back to $|1\rangle$.

We briefly highlight a few quantum query algorithms that we will invoke later.

2.2.1 Grover's algorithm

Consider the following *(unordered) search problem.* A database of size N is modeled as a binary string $x \in \{0,1\}^N$. A *solution* in the database is an index i such that $x_i = 1$. The goal of the search problem is to find a solution given query access to x. It is not hard to see that every classical algorithm that solves the search problem needs to make $\Omega(N)$ queries in the worst case. Grover [Gro96, BHMT02] came up with a quantum algorithm that finds a solution with high probability using $O(\sqrt{N})$ queries (this is also known to be optimal [BBBV97]).

For $N = 2^n$, let $D_n = 2|0^n\rangle\langle 0^n| - \mathrm{Id}$ be the unitary that puts '-1' in front of all basis states except $|0^n\rangle$. The *Grover iterate* $G = H^{\otimes n} D_n H^{\otimes n} O_{x,\pm}$ is a unitary that makes one quantum query. We now describe Grover's algorithm (assuming the number of solutions $|x| = 1$).

1. Start with $|0^n\rangle$.
2. Apply Hadamard transforms to all n qubits, obtaining $\frac{1}{\sqrt{N}} \sum_{i \in \{0,1\}^n} |i\rangle$.
3. Apply the Grover iterate G $\left\lceil \frac{\pi}{4}\sqrt{N} \right\rceil$ times.
4. Measure the final state to obtain an index $i \in [N]$.

One can show that with high probability the measurement outcome is a solution. If the number of solutions $|x| \geq 1$ is unknown, then a variant of this algorithm from [BHMT02] can be used to find a solution with high probability using $O(\sqrt{N/|x|})$ queries. We will later invoke the following more recent application of Grover's algorithm.

Theorem 2.1 ([Kot14],[LL16, Theorem 5.6])**.** *Suppose $x \in \{0,1\}^N$. There exists a quantum algorithm that satisfies the following properties:*

- *if $x \neq 0^N$, then let d be the first (i.e., smallest) index satisfying $x_d = 1$; the algorithm uses an expected number of $O(\sqrt{d})$ queries to x and outputs d with probability at least $2/3$;*
- *if $x = 0^N$ then the algorithm always outputs "no solution" after $O(\sqrt{N})$ queries.*

2.2.2 Fourier sampling

A very simple but powerful quantum algorithm is *Fourier sampling.* In order to explain it, let us first introduce the basics of Fourier analysis of functions on the Boolean cube (see [Wol08, O'D14] for more). Consider a function $f : \{0,1\}^n \to \mathbb{R}$. Its *Fourier coefficients* are $\widehat{f}(S) = \mathbb{E}_x[f(x)\chi_S(x)]$, where $S \in \{0,1\}^n$, the expectation is uniform over all $x \in \{0,1\}^n$, and $\chi_S(x) = (-1)^{x \cdot S}$ is the *character function* corresponding to S. The Fourier decomposition of f is $f = \sum_S \widehat{f}(S)\chi_S$. Parseval's identity says that $\sum_S \widehat{f}(S)^2 = \mathbb{E}_x[f(x)^2]$. Note that if f has range $\{\pm 1\}$ then Parseval implies that the squared Fourier coefficients $\widehat{f}(S)^2$ sum to 1, and hence form a probability distribution. Fourier sampling means sampling an S with probability $\widehat{f}(S)^2$. Classically this is a hard problem, because the probabilities depend on all 2^n values of f. However, the following quantum algorithm due to Bernstein and Vazirani [BV97] does it exactly using only 1 query and $O(n)$ gates.

1. Start with $|0^n\rangle$.

2. Apply Hadamard transforms to all n qubits, obtaining $\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$.

3. Query O_f,[6] obtaining $\frac{1}{\sqrt{2^n}} \sum_x f(x)|x\rangle$.

4. Apply Hadamard transforms to all n qubits to obtain
$$\frac{1}{\sqrt{2^n}} \sum_x f(x) \Big(\frac{1}{\sqrt{2^n}} \sum_S (-1)^{x\cdot S} |S\rangle \Big) = \sum_S \widehat{f}(S)|S\rangle.$$

5. Measure the state, obtaining S with probability $\widehat{f}(S)^2$.

2.3 Pretty Good Measurement

Consider an ensemble of m d-dimensional pure quantum states, $\mathcal{E} = \{(p_i, |\psi_i\rangle)\}_{i\in[m]}$, where $p_i \geq 0$ and $\sum_{i\in[m]} p_i = 1$. Suppose we are given an unknown state $|\psi_j\rangle$ sampled according to the probabilities $\{p_i\}$ and we are interested in maximizing the *average success probability* to identify the given state (i.e., to find j). For a POVM $\mathcal{M} = \{M_i\}_{i\in[m]}$, the average success probability is $P_{\mathcal{M}}(\mathcal{E}) = \sum_{i=1}^m p_i \langle\psi_i|M_i|\psi_i\rangle$.

Let $P^{opt}(\mathcal{E}) = \max_{\mathcal{M}} P_{\mathcal{M}}(\mathcal{E})$ denote the optimal average success probability of $\mathcal{E}$, maximized over the set of all m-outcome POVMs. The so-called *Pretty Good Measurement* (PGM) is a specific POVM (depending on $\mathcal{E}$), that does *reasonably* well against $\mathcal{E}$. We omit the details of the PGM and state only the crucial properties that we require. Suppose $P^{pgm}(\mathcal{E})$ is the average success probability in identifying the states in $\mathcal{E}$ using the PGM, then

$$P^{opt}(\mathcal{E}) \geq P^{pgm}(\mathcal{E}) \geq P^{opt}(\mathcal{E})^2,$$

where the second inequality was shown by Barnum and Knill [BK02]. For an ensemble $\mathcal{E} = \{(p_i, |\psi_i\rangle)\}_{i\in[m]}$, let $|\psi'_i\rangle = \sqrt{p_i}|\psi_i\rangle$ for $i \in [m]$. Let G be the $m \times m$ Gram matrix for $\{|\psi'_i\rangle\}_{i\in[m]}$, i.e., $G(i,j) = \langle\psi'_i|\psi'_j\rangle$ for $i, j \in [m]$. Then one can show that $P^{pgm}(\mathcal{E}) = \sum_{i\in[m]} \sqrt{G}(i,i)^2$ (see, e.g., [Mon07] or [AW16, Section 2.6]).

3 Learning models

In this section we will define the three main learning models that we focus on: the *exact* model of learning introduced by Angluin [Ang87], the *PAC* model of learning introduced by Valiant [Val84], and the *agnostic* model of learning introduced by Haussler [Hau92] and Kearns et al. [KSS94].

Below, a *concept class* $\mathcal{C}$ will usually be a set of functions $c : \{0,1\}^n \to \{0,1\}$, though we can also allow functions $c : [N] \to \{0,1\}$, or treat such a c as an N-bit string specified by its truth-table.

[6] Here we view $f \in \{1,-1\}^{2^n}$ as being specified by its truth-table.

3.1 Exact learning

Classical exact learning. In the exact learning model, a learner $\mathcal{A}$ for a concept class $\mathcal{C}$ is given access to a *membership oracle* $\mathrm{MQ}(c)$ for the *target concept* $c \in \mathcal{C}$ that $\mathcal{A}$ is trying to learn. Given an input $x \in \{0,1\}^n$, $\mathrm{MQ}(c)$ returns the label $c(x)$. A learning algorithm $\mathcal{A}$ is an *exact learner* for $\mathcal{C}$ if:

> For every $c \in \mathcal{C}$, given access to the $\mathrm{MQ}(c)$ oracle:
> with probability at least $2/3$, $\mathcal{A}$ outputs an h such that $h(x) = c(x)$ for all $x \in \{0,1\}^n$.[7]

This model is also sometimes known as "oracle identification": the idea is that $\mathcal{C}$ is a set of possible oracles, and we want to efficiently identify which $c \in \mathcal{C}$ is our actual oracle, using membership queries to c.

The *query complexity* of $\mathcal{A}$ is the maximum number of invocations of the $\mathrm{MQ}(c)$ oracle which the learner makes, over all concepts $c \in \mathcal{C}$ and over the internal randomness of the learner. The *query complexity of exactly learning* $\mathcal{C}$ is the minimum query complexity over all exact learners for $\mathcal{C}$.[8]

Each concept $c : \{0,1\}^n \to \{0,1\}$ can also be specified by its N-bit truth-table (with $N = 2^n$), hence one may view the concept class $\mathcal{C}$ as a subset of $\{0,1\}^N$. For a given N and M, define the (N,M)*-query complexity of exact learning* as the maximum query complexity of exactly learning $\mathcal{C}$, maximized over all $\mathcal{C} \subseteq \{0,1\}^N$ such that $|\mathcal{C}| = M$.

Quantum exact learning. In the quantum setting, instead of having access to an $\mathrm{MQ}(c)$ oracle, a *quantum exact learner* is given access to a $\mathrm{QMQ}(c)$ oracle, which corresponds to the map $\mathrm{QMQ}(c) : |x,b\rangle \to |x, b \oplus c(x)\rangle$ for $x \in \{0,1\}^n, b \in \{0,1\}$. For a given $\mathcal{C}, N, M$, one can define the *quantum query complexity of exactly learning* $\mathcal{C}$, and the (N,M)*-quantum query complexity of exact learning* as the quantum analogues to the classical complexity measures.

3.2 Probably Approximately Correct (PAC) learning

Classical PAC model. In the PAC model, a learner $\mathcal{A}$ is given access to a *random example oracle* $\mathrm{PEX}(c,D)$, where $c \in \mathcal{C}$ is the *target concept* that $\mathcal{A}$ is trying to learn and $D : \{0,1\}^n \to [0,1]$ is an unknown probability distribution. When invoked, $\mathrm{PEX}(c,D)$ returns a labeled example $(x, c(x))$ where x is drawn from D. A learning algorithm $\mathcal{A}$ is an (ε,δ)*-PAC learner* for $\mathcal{C}$ if:

> For every $c \in \mathcal{C}$ and distribution D, given access to the $\mathrm{PEX}(c,D)$ oracle:
> with probability at least $1-\delta$, $\mathcal{A}$ outputs an h such that $\Pr_{x\sim D}[h(x) \neq c(x)] \leq \varepsilon$.

Note that the learner has the freedom to output an hypothesis h which is not itself in the concept class $\mathcal{C}$. If the learner always produces an $h \in \mathcal{C}$, then it is called a *proper* PAC learner.

The *sample complexity* of $\mathcal{A}$ is the maximum number of invocations of the $\mathrm{PEX}(c,D)$ oracle which the learner makes, over all concepts $c \in \mathcal{C}$, distributions D, and the internal randomness of the learner. The (ε,δ)*-PAC sample complexity* of a concept class $\mathcal{C}$ is the minimum sample complexity over all (ε,δ)-PAC learners for $\mathcal{C}$.

[7] We could also consider a δ-exact learner who succeeds with probability $1-\delta$, but here restrict to $\delta = 1/3$ for simplicity. Standard amplification techniques can reduce this $1/3$ to any $\delta > 0$ at the expense of an $O(\log(1/\delta))$ factor in the complexity.

[8] This terminology of "learning $\mathcal{C}$" or "$\mathcal{C}$ is learnable" is fairly settled though slightly unfortunate: what is actually being learned is of course a target concept $c \in \mathcal{C}$, not the class $\mathcal{C}$ itself, which the learner already knows from the start.

Quantum PAC model. The quantum PAC model was introduced by Bshouty and Jackson [BJ99]. Instead of having access to a $\mathrm{PEX}(c, D)$ oracle, the *quantum PAC learner* has access to a *quantum example oracle* $\mathrm{QPEX}(c, D)$ that produces a *quantum example*

$$\sum_{x \in \{0,1\}^n} \sqrt{D(x)} |x, c(x)\rangle.$$

Such a quantum example is the natural quantum generalization of a classical random sample.[9] While it is not always realistic to assume access to such (fragile) quantum states, one can certainly envision learning situations where the data is provided by a coherent quantum process.

A quantum PAC learner is given access to several copies of the quantum example and performs a POVM, where each outcome is associated with an hypothesis. Its *sample complexity* is the maximum number of invocations of the $\mathrm{QPEX}(c, D)$ oracle which the learner makes, over all distributions D and over the learner's internal randomness. We define the (ε, δ)-*quantum PAC sample complexity* of $\mathcal{C}$ as the minimum sample complexity over all (ε, δ)-quantum PAC learners for $\mathcal{C}$.

Observe that from a quantum example $\sum_x \sqrt{D(x)} |x, c(x)\rangle$, we can obtain $\sum_x \sqrt{D(x)} (-1)^{c(x)} |x\rangle$ with probability $1/2$: apply the Hadamard transform to the last qubit and measure it. With probability $1/2$ we obtain the outcome 1, in which case the remaining state is $\sum_x \sqrt{D(x)} (-1)^{c(x)} |x\rangle$. If D is the uniform distribution, then the obtained state is exactly the state needed in step 3 of the Fourier sampling algorithm described in Section 2.2.2.

How does the model of quantum examples compare to the model of quantum membership queries? If the distribution D is known, a membership query can be used to create a quantum example: the learner can create the superposition $\sum_x \sqrt{D(x)} |x, 0\rangle$ and apply a membership query to the target concept c to obtain a quantum example. On the other hand, as Bshouty and Jackson [BJ99] already observed, a membership query cannot be simulated using a small number of quantum examples. Consider for example the learning problem corresponding to Grover search, where the concept class $\mathcal{C} \subseteq \{0,1\}^N$ consists of all strings of weight 1. We know that $\Theta(\sqrt{N})$ quantum membership queries are necessary and sufficient to exactly learn the target concept with high probability. However, it is not hard to show that, under the uniform distribution, one needs $\Omega(N)$ quantum examples to exactly learn the target concept with high probability. Hence simulating one membership query requires at least $\Omega(\sqrt{N})$ quantum examples.

3.3 Agnostic learning

Classical agnostic model. In the PAC model one assumes that the labeled examples are generated perfectly according to a target concept $c \in \mathcal{C}$, which is often not a realistic assumption. In the agnostic model, for an unknown distribution $D : \{0,1\}^{n+1} \to [0,1]$, the learner is given access to an $\mathrm{AEX}(D)$ oracle. Each invocation of $\mathrm{AEX}(D)$ produces labeled examples (x, b) drawn from the distribution D (where $x \in \{0,1\}^n$ and $b \in \{0,1\}$). Define the error of $h : \{0,1\}^n \to \{0,1\}$ under D as $\mathrm{err}_D(h) = \Pr_{(x,b)\sim D}[h(x) \neq b]$. When h is restricted to come from a concept class $\mathcal{C}$, the minimal error achievable is $\mathrm{opt}_D(\mathcal{C}) = \min_{c \in \mathcal{C}}\{\mathrm{err}_D(c)\}$. A learning algorithm $\mathcal{A}$ is an (ε, δ)-*agnostic learner* for $\mathcal{C}$ if it can produce a hypothesis $h \in \mathcal{C}$ whose error is not much worse:

> For every distribution D on $\{0,1\}^{n+1}$, given access to the $\mathrm{AEX}(D)$ oracle:
> with probability at least $1 - \delta$, $\mathcal{A}$ outputs an $h \in \mathcal{C}$ such that $\mathrm{err}_D(h) \leq \mathrm{opt}_D(\mathcal{C}) + \varepsilon$.

[9] We could also allow complex phases for the amplitudes $\sqrt{D(x)}$; however, these will make no difference for the results presented here.

If there exists a $c \in \mathcal{C}$ that perfectly classifies every x with label b for all (x, b) such that $D(x, b) > 0$, then $\mathrm{opt}_D(\mathcal{C}) = 0$ and we are in the setting of proper PAC learning. The *sample complexity* of $\mathcal{A}$ is the maximum number of invocations of the $\mathrm{AEX}(D)$ oracle which the learner makes, over all distributions D and over the learner's internal randomness. The (ε, δ)-*agnostic sample complexity* of a concept class $\mathcal{C}$ is the minimum sample complexity over all (ε, δ)-agnostic learners for $\mathcal{C}$.

Quantum agnostic model. The model of quantum agnostic learning was first studied in [AW16]. For a distribution $D : \{0,1\}^{n+1} \to [0,1]$, the *quantum agnostic learner* has access to a $\mathrm{QAEX}(D)$ oracle that produces a quantum example $\sum_{(x,b)\in\{0,1\}^{n+1}} \sqrt{D(x,b)}|x,b\rangle$. A quantum agnostic learner is given access to several copies of the quantum example and performs a POVM at the end. Similar to the classical complexities, one can define (ε, δ)-*quantum agnostic sample complexity* as the minimum sample complexity over all (ε, δ)-quantum agnostic learners for $\mathcal{C}$.

4 Results on query complexity and sample complexity

4.1 Query complexity of exact learning

In this section, we begin by proving bounds on the quantum query complexity of exactly learning a concept class $\mathcal{C}$ in terms of a combinatorial parameter $\gamma(\mathcal{C})$, which we define shortly, and then sketch the proof of optimal bounds on (N, M)-quantum query complexity of exact learning.

Throughout this section, we will specify a concept $c : \{0,1\}^n \to \{0,1\}$ by its N-bit truth-table (with $N = 2^n$), hence $\mathcal{C} \subseteq \{0,1\}^N$. For a set $S \subseteq \{0,1\}^N$, we will often use the "N-bit majority string" $\mathrm{MAJ}(S) \in \{0,1\}^N$ defined as: $\mathrm{MAJ}(S)_i = 1$ iff $|\{s \in S : s_i = 1\}| \geq |\{s \in S : s_i = 0\}|$.

Definition 4.1. *(Combinatorial parameter $\gamma(\mathcal{C})$) Fix a concept class $\mathcal{C} \subseteq \{0,1\}^N$ and let $\mathcal{C}' \subseteq \mathcal{C}$. For $i \in [N]$ and $b \in \{0,1\}$, define*

$$\gamma(\mathcal{C}', i, b) = \frac{|\{c \in \mathcal{C}' : c_i = b\}|}{|\mathcal{C}'|}$$

as the fraction of concepts in $\mathcal{C}'$ that satisfy $c_i = b$. Let $\gamma(\mathcal{C}', i) = \min\{\gamma(\mathcal{C}', i, 0), \gamma(\mathcal{C}', i, 1)\}$ be the minimum fraction of concepts that can be eliminated by learning c_i. Let

$$\gamma(\mathcal{C}') = \max_{i\in[N]}\{\gamma(\mathcal{C}', i)\}$$

denote the largest fraction of concepts in $\mathcal{C}'$ that can be eliminated by a query. Finally, define

$$\gamma(\mathcal{C}) = \min_{\substack{\mathcal{C}'\subseteq\mathcal{C},\\ |\mathcal{C}'|\geq 2}} \gamma(\mathcal{C}') = \min_{\substack{\mathcal{C}'\subseteq\mathcal{C},\\ |\mathcal{C}'|\geq 2}} \max_{i\in[N]} \min_{b\in\{0,1\}} \gamma(\mathcal{C}', i, b).$$

This complicated-looking definition is motivated by the following learning algorithm. Suppose the learner wants to exactly learn $c \in \mathcal{C}$. Greedily, the learner would query c on the "best" input $i \in [N]$, i.e., the i that eliminates the largest fraction of concepts from $\mathcal{C}$ irrespective of the value of c_i. Suppose j is the "best" input (i.e., $i = j$ maximizes $\gamma(\mathcal{C}, i)$) and the learner queries c on index j: at least a $\gamma(\mathcal{C})$-fraction of the concepts in $\mathcal{C}$ will be inconsistent with the query-outcome, and these can now be eliminated from $\mathcal{C}$. Call the set of remaining concepts $\mathcal{C}'$, and note that $|\mathcal{C}'| \leq (1 - \gamma(\mathcal{C}))|\mathcal{C}|$. The outermost min in $\gamma(\mathcal{C})$ guarantees that there will be another query that the learner can make to eliminate at least a $\gamma(\mathcal{C})$-fraction of the remaining concepts from $\mathcal{C}'$, and so on. We stop when there is only one remaining concept left. Since each query will shrink the set of remaining concepts by a factor of at least $(1-\gamma(\mathcal{C}))^T$, making $T = O((\log|\mathcal{C}|)/\gamma(\mathcal{C}))$ queries suffices to shrink $\mathcal{C}$ to $\{c\}$.

4.1.1 Query complexity of exactly learning $\mathcal{C}$ in terms of $\gamma(\mathcal{C})$

Bshouty et al. [BCG$^+$96] showed the following bounds on the classical complexity of exactly learning a concept class $\mathcal{C}$ (we already sketched the upper bound above).

Theorem 4.2 ([BCG$^+$96, SG04])**.** *Every classical exact learner for concept class $\mathcal{C}$ has to use $\Omega(\max\{1/\gamma(\mathcal{C}), \log|\mathcal{C}|\})$ membership queries. For every $\mathcal{C}$, there is a classical exact learner which learns $\mathcal{C}$ using $O(\frac{\log|\mathcal{C}|}{\gamma(\mathcal{C})})$ membership queries.*

In order to show a polynomial relation between quantum and classical exact learning, Servedio and Gortler [SG04] showed the following lower bounds.

Theorem 4.3 ([SG04])**.** *Let $N = 2^n$. Every quantum exact learner for concept class $\mathcal{C} \subseteq \{0,1\}^N$ has to make $\Omega(\max\{\frac{1}{\sqrt{\gamma(\mathcal{C})}}, \frac{\log|\mathcal{C}|}{n}\})$ membership queries.*

Proof sketch. We first prove the $\Omega(1/\sqrt{\gamma(\mathcal{C})})$ lower bound. We will use the unweighted adversary bound of Ambainis [Amb02]. One version of this bound says the following: suppose we have a quantum algorithm with possible inputs $\mathcal{D} \subseteq \{0,1\}^N$, and a relation $R \subseteq \mathcal{D} \times \mathcal{D}$ (equivalently, a bipartite graph) with the following properties:

1. Every left-vertex v is related to at least m right-vertices w (i.e., $|\{w \in \mathcal{D} : (v,w) \in R\}| \geq m$).

2. Every right-vertex w is related to at least m' left-vertices v (i.e., $|\{v \in \mathcal{D} : (v,w) \in R\}| \geq m'$).

3. For every $i \in [N]$, every left-vertex v is related to at most ℓ right-vertices w satisfying $v_i \neq w_i$.

4. For every $i \in [N]$, every right-vertex w is related to at most ℓ' left-vertices v satisfying $v_i \neq w_i$.

Suppose that for every $(v,w) \in R$, the final states of our algorithm on inputs v and w are $\Omega(1)$ apart in trace norm. Then the quantum algorithm makes $\Omega(\sqrt{mm'/\ell\ell'})$ queries.

Now we want to apply this lower bound to a quantum exact learner for concept class $\mathcal{C}$. We can think of the learning algorithm as making queries to an N-bit input string and producing the name of a concept $c \in \mathcal{C}$ as output. Suppose $\mathcal{C}' \subseteq \mathcal{C}$ minimizes $\gamma(\mathcal{C})$ (i.e., $\gamma(\mathcal{C}') = \gamma(\mathcal{C})$). Define $\tilde{c} = \mathrm{MAJ}(\mathcal{C}')$. For every $i \in [N]$, by the definition of $\gamma(\mathcal{C}', i)$, $\tilde{c}$ disagrees on i with at most a $\gamma(\mathcal{C}', i)$-fraction of the $c \in \mathcal{C}'$. Let $\mathcal{D} = \mathcal{C}' \cup \{\tilde{c}\}$. Note that $\tilde{c}$ need not be in $\mathcal{C}'$ or even in $\mathcal{C}$, but we can still consider what our learner does on input $\tilde{c}$. We consider two cases:

Case 1: For every $c \in \mathcal{C}'$, the probability that the learner outputs c when run on the typical concept $\tilde{c}$, is $< 1/2$. In this case we pick our relation $R = \{\tilde{c}\} \times \mathcal{C}'$. Calculating the parameters for the adversary bound, we have $m = |\mathcal{C}'|$, $m' = 1$, $\ell \leq \gamma(\mathcal{C}')|\mathcal{C}'|$ (because for every i, $\tilde{c}_i \neq c_i$ for at most a $\gamma(\mathcal{C}', i)$-fraction of the $c \in \mathcal{C}'$ and $\gamma(\mathcal{C}', i) \leq \gamma(\mathcal{C}')$ by definition), and $\ell' = 1$. Since, for every $c \in \mathcal{C}'$, the learner outputs c with high probability on input c, the final states on every pair of R-related concepts will be $\Omega(1)$ apart. Hence, the number of queries that our learner makes is $\Omega(\sqrt{mm'/\ell\ell'}) = \Omega(1/\sqrt{\gamma(\mathcal{C}')}) = \Omega(1/\sqrt{\gamma(\mathcal{C})})$ (because $\mathcal{C}'$ minimized $\gamma(\mathcal{C})$).

Case 2: There exists a specific $c \in \mathcal{C}'$ that the learner gives as output with probability $\geq 1/2$ when run on input $\tilde{c}$. In this case we pick $R = \{\tilde{c}\} \times (\mathcal{C}' \backslash \{c\})$, ensuring that the final states on every pair of R-related concepts will be $\Omega(1)$ apart. We now have $m = |\mathcal{C}'| - 1$, $m' = 1$, $\ell \leq \gamma(\mathcal{C}')|\mathcal{C}'|$, and $\ell' = 1$. Since $(|\mathcal{C}'| - 1)/|\mathcal{C}'| = \Omega(1)$, the adversary bound again yields a $\Omega(1/\sqrt{\gamma(\mathcal{C})})$ bound.

We now prove the $\Omega((\log |\mathcal{C}|)/n)$ lower bound by an information-theoretic argument, as follows. View the target string $c \in \mathcal{C}$ as a uniformly distributed random variable. If our algorithm can exactly identify c with high success probability, it has learned $\Omega(\log |\mathcal{C}|)$ bits of information about c (formally, the mutual information between c and the learner's output is $\Omega(\log |\mathcal{C}|)$). From Holevo's theorem [Hol73], since a quantum query acts on only $n+1$ qubits, one quantum query can yield at most $O(n)$ bits of information about c. Hence $\Omega((\log |\mathcal{C}|)/n)$ quantum queries are needed. □

Both of the above lower bounds are in fact individually optimal. First, if one takes $\mathcal{C} \subseteq \{0,1\}^N$ to consist of the N functions c for which $c(i) = 1$ for exactly one i, then exact learning corresponds to the unordered search problem with 1 solution. Here $\gamma(\mathcal{C}) = 1/N$, and $\Theta(\sqrt{N})$ queries are necessary and sufficient thanks to Grover's algorithm. Second, if $\mathcal{C}$ is the class of $N = 2^n$ linear functions on $\{0,1\}^n$, $\mathcal{C} = \{c(x) = a \cdot x : a \in \{0,1\}^n\}$, then Fourier sampling gives an $O(1)$-query algorithm (see Section 5.3). In addition to these quantum-classical separations based on Grover and Fourier sampling, in Section 5.3 we also mention a fourth-power separation between $Q(\mathcal{C})$ and $D(\mathcal{C})$ due to Belovs [Bel15], for the problem of learning certain *k-juntas*.

Combining Theorems 4.2 and 4.3, Servedio and Gortler [SG04] showed that the classical and quantum query complexity of exact learning are essentially polynomially related for every $\mathcal{C}$.

Corollary 4.4 ([SG04])**.** *If concept class $\mathcal{C}$ has classical and quantum membership query complexities $D(\mathcal{C})$ and $Q(\mathcal{C})$, respectively, then $D(\mathcal{C}) = O(nQ(\mathcal{C})^3)$.*

4.1.2 (N, M)-query complexity of exact learning

In this section we focus on the (N, M)-quantum query complexity of exact learning. Classically, the following characterization is easy to prove.

Theorem 4.5 (Folklore)**.** *The (N, M)-query complexity of exact learning is $\Theta(\min\{M, N\})$.*

In the quantum context, the (N, M)-query complexity of exact learning has been completely characterized by Kothari [Kot14]. Improving on [AIK$^+$04, AIK$^+$07, AIN$^+$09], he showed the following theorem.

Theorem 4.6 ([Kot14])**.** *The (N, M)-quantum query complexity of exact learning is $\Theta(\sqrt{M})$ for $M \leq N$ and $\Theta\left(\sqrt{\frac{N \log M}{\log(N/\log M)+1}}\right)$ for $N < M \leq 2^N$.*

Proof sketch. Consider first the lower bound for the case $M \leq N$. Suppose $\mathcal{C} \subseteq \{c \in \{0,1\}^N : |c| = 1\}$ satisfies $|\mathcal{C}| = M$. Then, exactly learning $\mathcal{C}$ is as hard as the unordered search problem on M bits, which requires $\Omega(\sqrt{M})$ quantum queries. The lower bound for the case $N < M \leq 2^N$ is fairly technical and we refer the reader to [AIN$^+$09].

We now sketch the proofs of the upper bound. We use the following notation: for $u \in \{0,1\}^n$ and $S \subseteq [n]$, let $u_S \in \{0,1\}^{|S|}$ be the string obtained by restricting u to the indices in S.

We first describe a quantum algorithm that gives a worse upper bound than promised, but is easy to explain. Suppose $\mathcal{C} \subseteq \{0,1\}^N$ satisfies $|\mathcal{C}| = M$. Let $c \in \mathcal{C}$ be the unknown target concept that the algorithm is trying to learn. The basic idea of the algorithm is as follows: use the algorithm of Theorem 2.1 to find the first index $p_1 \in [N]$ at which c and $\mathrm{MAJ}(\mathcal{C})$ differ. This uses an expected $O(\sqrt{p_1})$ queries to c (if there is no difference, i.e., $c = \mathrm{MAJ}(\mathcal{C})$, then the algorithm will tell us so after $O(\sqrt{N})$ queries and we can stop). We have now learned the first p_1 bits of c. Let $\mathcal{C}_1 = \{z_{[N]\setminus[p_1]} : z \in \mathcal{C},\ z_{[p_1-1]} = \mathrm{MAJ}(\mathcal{C})_{[p_1-1]},\ z_{p_1} = \overline{\mathrm{MAJ}(\mathcal{C})}_{p_1}\} \subseteq \{0,1\}^{N-p_1}$ be the set of suffixes of the concepts in $\mathcal{C}$ that agree with $\mathrm{MAJ}(\mathcal{C})$ on the first $p_1 - 1$ indices and disagree with $\mathrm{MAJ}(\mathcal{C})$ on the p_1th index. Similarly, let $c^1 = c_{[N]\setminus[p_1]}$ be the "updated" unknown target concept after restricting c to the coordinates $\{p_1+1, \ldots, N\}$. Next, we use the same idea to find the first index $p_2 \in [N - p_1]$ such that $(c^1)_{p_2} \neq \mathrm{MAJ}(\mathcal{C}_1)_{p_2}$. Repeat this until only one concept is left, and let r be the number of repetitions (i.e., until $|\mathcal{C}_r| = 1$).

In order to analyze the query complexity, first note that, for $k \geq 1$, the k-th iteration of the procedure gives us p_k bits of c. Since the procedure repeated r times, we have $p_1 + \cdots + p_r \leq N$. Second, each repetition in the algorithm reduces the size of $\mathcal{C}_i$ by at least a half, i.e., for $i \geq 2$, $|\mathcal{C}_i| \leq |\mathcal{C}_{i-1}|/2$. Hence one needs to repeat the procedure at most $r \leq O(\log M)$ times. The last run will use $O(\sqrt{N})$ queries and will tell us that we have learned all the bits of c. It follows that the total number of queries the algorithm makes to c is

$$\sum_{k=1}^{r} O(\sqrt{p_k}) + O(\sqrt{N}) \leq O\left(\sqrt{r}\sqrt{\sum_{k=1}^{r} p_k}\right) + O(\sqrt{N}) \leq O(\sqrt{N \log M}),$$

where we used the Cauchy-Schwarz inequality and our upper bounds on r and $\sum_i p_i$.[10]

This algorithm is an $O(\sqrt{\log(N/\log M)})$-factor away from the promised upper bound. Tweaking the algorithm to save the logarithmic factor uses the following lemma by [Heg95]. It shows that there exists an explicit ordering and a string s^i such that replacing $\mathrm{MAJ}(\mathcal{C}_i)$ in the basic algorithm leads to faster reduction of $|\mathcal{C}_i|$.

Lemma 4.7 ([Heg95, Lemma 3.2])**.** *Let $L \in \mathbb{N}$ and $\mathcal{C} \subseteq \{0,1\}^L$. There exists $s \in \{0,1\}^L$ and permutation $\pi : [L] \to [L]$, such that for every $p \in [L]$, we have $|\mathcal{C}_p| \leq \frac{|\mathcal{C}|}{\max\{2,p\}}$, where $\mathcal{C}_p = \{c \in \mathcal{C} : c_{\{\pi(1),\ldots,\pi(p-1)\}} = s_{\{\pi(1),\ldots,\pi(p-1)\}},\ c_{\pi(p)} \neq s_{\pi(p)}\}$ is the set of strings in $\mathcal{C}$ that agree with s at $\pi(1), \ldots, \pi(p-1)$ and disagree at $\pi(p)$.*

We now describe the final algorithm.

1. Set $\mathcal{C}_1 := \mathcal{C}$, $N_1 := N$, and $c^1 := c$.

2. Repeat until $|\mathcal{C}_k| = 1$

    - Let $s^k \in \{0,1\}^{N_k}$ be the string and $\pi^k : [N_k] \to [N_k]$ be the permutation obtained by applying Lemma 4.7 to $\mathcal{C}_k$ (with $L = N_k$)
    - Search for the first (according to π^k) disagreement between s^k and c^k using the algorithm of Theorem 2.1. Suppose we find a disagreement at index $\pi^k(p_k) \in [N_k]$, i.e., s^k and c^k agree on the indices $I_k = \{\pi^k(1), \ldots, \pi^k(p_k - 1)\}$

[10] One has to be careful here because each run of the algorithm of Theorem 2.1 has a small error probability. Kothari shows how this can be handled *without* the super-constant blow-up in the overall complexity that would follow from naive error reduction.

- Set $N_{k+1} := N_k - p_k$, $c^{k+1} := c^k_{[N_k]\setminus(I_k\cup\{\pi^k(p_k)\})}$ and
$\mathcal{C}_{k+1} := \{u_{[N_k]\setminus(I_k\cup\{\pi^k(p_k)\})} : u \in \mathcal{C}_k,\ u_{I_k} = s^k_{I_k},\ u_{\pi^k(p_k)} \neq s^k_{\pi^k(p_k)}\}$

3. Output the unique element of $\mathcal{C}_k$.

Let r be the number of times the loop in Step 2 repeats and suppose in the k-th iteration we learned p_k bits of c. Then we have $\sum_{k=1}^r p_k \leq N$. The overall query complexity is $T = O(\sum_{k=1}^r \sqrt{p_k})$. Earlier we had $|\mathcal{C}_{k+1}| \leq |\mathcal{C}_k|/2$ and hence $r \leq O(\log M)$. But now, from Lemma 4.7 we have $|\mathcal{C}_{k+1}| \leq |\mathcal{C}_k|/\max\{2, p_k\}$. Since each iteration reduces the size of $\mathcal{C}_k$ by a factor of $\max\{2, p_k\}$, we have $\prod_{k=1}^r \max\{2, p_k\} \leq M$. Solving this optimization problem (i.e., $\min T$ s.t. $\prod_{k=1}^r \max\{2, p_k\} \leq M$, $\sum_{k=1}^r p_k \leq N$), Kothari showed

$$T = O(\sqrt{M}) \quad \text{if } M \leq N, \quad \text{and } T = O\Big(\sqrt{\frac{N \log M}{\log(N/\log M) + 1}}\Big) \quad \text{if } M > N. \qquad \square$$

Kothari [Kot14], improving upon [SG04, AS05], resolved a conjecture of Hunziker et al. [HMP$^+$10] by showing the following upper bound for quantum query complexity of exact learning. This is exactly the above algorithm, analyzed in terms of $\gamma(\mathcal{C})$.

Theorem 4.8 ([Kot14]). *For every concept class $\mathcal{C}$, there is a quantum exact learner for $\mathcal{C}$ using $O\Big(\sqrt{\frac{1/\gamma(\mathcal{C})}{\log(1/\gamma(\mathcal{C}))}} \log|\mathcal{C}|\Big)$ quantum membership queries.*

Moshkin [Mos83] introduced another combinatorial parameter, which Hegedűs [Heg95] called the *extended teaching dimension EXT-$TD(\mathcal{C})$* of a concept class $\mathcal{C}$ (we shall not define EXT-$TD(\mathcal{C})$ here, see [Heg95] for a precise definition). Building upon the work of [Mos83], Hegedűs proved the following theorem.

Theorem 4.9 ([Mos83],[Heg95, Theorem 3.1]). *Every classical exact learner for concept class $\mathcal{C}$ has to use $\Omega(\max\{EXT\text{-}TD(\mathcal{C}), \log|\mathcal{C}|\})$ membership queries. For every $\mathcal{C}$, there is a classical exact learner which learns $\mathcal{C}$ using $O\left(\frac{EXT\text{-}TD(\mathcal{C})}{\log(EXT\text{-}TD(\mathcal{C}))} \log|\mathcal{C}|\right)$ membership queries.*

Comparing this with Theorem 4.2, observe that both $1/\gamma(\mathcal{C})$ and EXT-$TD(\mathcal{C})$ give lower bounds on classical query complexity, but the upper bound in terms of EXT-$TD(\mathcal{C})$ is better by a logarithmic factor. Also for analyzing quantum complexity, EXT-$TD(\mathcal{C})$ may be a superior parameter.

4.2 Sample complexity of PAC learning

One of the most fundamental results in learning theory is that the sample complexity of $\mathcal{C}$ is tightly determined by a combinatorial parameter called the *VC dimension* of $\mathcal{C}$, named after Vapnik and Chervonenkis [VC71] and defined as follows.

Definition 4.10. *(VC dimension) Fix a concept class $\mathcal{C}$ over $\{0,1\}^n$. A set $\mathcal{S} = \{s_1, \ldots, s_t\} \subseteq \{0,1\}^n$ is said to be* shattered *by a concept class $\mathcal{C}$ if $\{(c(s_1)\cdots c(s_t)) : c \in \mathcal{C}\} = \{0,1\}^t$. In other words, for every labeling $\ell \in \{0,1\}^t$, there exists a $c \in \mathcal{C}$ such that $(c(s_1)\cdots c(s_t)) = \ell$. The VC dimension of $\mathcal{C}$ is the size of a largest $\mathcal{S} \subseteq \{0,1\}^n$ that is shattered by $\mathcal{C}$.*

Blumer et al. [BEHW89] proved that the (ε, δ)-PAC sample complexity of a concept class $\mathcal{C}$ with VC dimension d, is lower bounded by $\Omega(d/\varepsilon + \log(1/\delta)/\varepsilon)$, and they proved an upper bound that was worse by only a $\log(1/\varepsilon)$-factor. In recent work, Hanneke [Han16] (improving on Simon [Sim15]) got rid of this logarithmic factor, showing that the lower bound of Blumer et al. is in fact optimal. Combining these bounds, we have the following theorem.

Theorem 4.11 ([BEHW89, Han16]). *Let $\mathcal{C}$ be a concept class with VC-dim$(\mathcal{C}) = d + 1$. Then, $\Theta\Big(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\Big)$ examples are necessary and sufficient for an (ε, δ)-PAC learner for $\mathcal{C}$.*

This characterizes the number of samples necessary and sufficient for a classical PAC learning in terms of the VC dimension. How many *quantum* examples are needed to learn a concept class $\mathcal{C}$ of VC dimension d? Trivially, *upper* bounds on classical sample complexity imply upper bounds on quantum sample complexity. For some fixed distributions, in particular the uniform one, we will see in the next section that quantum examples can be more powerful than classical examples.

However, PAC learning requires a learner to be able to learn c under *all possible* distributions D, not just uniform. We showed that quantum examples are *not more powerful* than classical examples in the PAC model, improving over the results of [AS05, Zha10].

Theorem 4.12 ([AW16]). *Let $\mathcal{C}$ be a concept class with VC-dim$(\mathcal{C}) = d + 1$. Then, for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/20)$, $\Omega\Big(\frac{d}{\varepsilon} + \frac{1}{\varepsilon}\log\frac{1}{\delta}\Big)$ examples are necessary for an (ε, δ)-quantum PAC learner for $\mathcal{C}$.*

Proof sketch. The d-independent part of the lower bound has an easy proof, which we omit. In order to prove the $\Omega(d/\varepsilon)$ bound, we first define a distribution D on the shattered set $\mathcal{S} = \{s_0, \ldots, s_d\} \subseteq \{0,1\}^n$ as follows: $D(s_0) = 1 - 20\varepsilon$ and $D(s_i) = 20\varepsilon/d$ for all $i \in [d]$.

A quantum PAC learner is given T copies of the quantum example for an unknown concept c and needs to output a hypothesis h that is ε-close to c. We want to relate this to the state identification problem of Section 2.3. In order to render ε-approximation of c equivalent to *identification* of c, we use a $[d, k, r]_2$ linear error-correcting code for $k \geq d/4$, distance $r \geq d/8$, with generator matrix $M \in \mathbb{F}_2^{d \times k}$ (we know such codes exist if d is a sufficiently large constant). Let $\{Mz : z \in \{0,1\}^k\} \subseteq \{0,1\}^d$ be the set of 2^k codewords in this linear code; these have Hamming distance $d_H(Mz, My) \geq d/8$ whenever $z \neq y$. For each $z \in \{0,1\}^k$, consider a concept c^z defined on the shattered set as: $c^z(s_0) = 0$ and $c^z(s_i) = (Mz)_i$ for all $i \in [d]$. Such concepts exist in $\mathcal{C}$ because $\mathcal{S}$ is shattered by $\mathcal{C}$. Additionally, since $r \geq d/8$ we have $\Pr_{s \sim D}[c^z(s) \neq c^y(s)] \geq 5\varepsilon/2$ whenever $z \neq y$. Hence, with probability at least $1 - \delta$, an (ε, δ)-PAC quantum learner trying to ε-approximate a concept from $\{c^z : z \in \{0,1\}^k\}$ will exactly *identify* the concept.

Consider the following state identification problem: for $z \in \{0,1\}^k$ let $|\psi_z\rangle = \sum_{i \in \{0,\ldots,d\}} \sqrt{D(s_i)}|s_i, c^z(s_i)\rangle$, and $\mathcal{E} = \{(2^{-k}, |\psi_z\rangle^{\otimes T})\}_{z \in \{0,1\}^k}$. Let G be the $2^k \times 2^k$ Gram matrix for this $\mathcal{E}$. From Section 2.3, we know that the average success probability of the PGM is $\sum_{z \in \{0,1\}^k} \sqrt{G}(z,z)^2$. Before we compute $\sqrt{G}(z,z)$, note that the (z,y)-th entry of G is a function of $z \oplus y$:

$$G(z, y) = \frac{1}{2^k}\langle\psi_z|\psi_y\rangle^T = \frac{1}{2^k}\Big(1 - \frac{20\varepsilon}{d}|M(z \oplus y)|\Big)^T.$$

The following claim will be helpful in analyzing the $\sqrt{G}(z,z)$ entry of the Gram matrix.

Theorem 4.13 ([AW16, Theorem 17])**.** *For $m \geq 10$, let $f : \{0,1\}^m \to \mathbb{R}$ be defined as $f(w) = (1 - \beta\frac{|w|}{m})^T$ for some $\beta \in (0,1]$ and $T \in [1, m/(e^3\beta)]$. For $k \leq m$, let $M \in \mathbb{F}_2^{m \times k}$ be a matrix with rank k. Suppose matrix $A \in \mathbb{R}^{2^k \times 2^k}$ is defined as $A(z,y) = (f \circ M)(z \oplus y)$ for $z, y \in \{0,1\}^k$, then*

$$\sqrt{A}(z,z) \leq e^{O(T^2\beta^2/m + \sqrt{Tm\beta})} \qquad \text{for all } z \in \{0,1\}^k.$$

We will not prove this, but mention that the proof of the theorem crucially uses the fact that the (z,y)-entry of matrix A is a function of $z \oplus y$, which allows us to diagonalize A easily. Using the theorem and the definition of $P^{pgm}(\mathcal{E})$ from Section 2.3, we have

$$P^{pgm}(\mathcal{E}) = \sum_{z \in \{0,1\}^k} \sqrt{G}(z,z)^2 \overset{\text{Thm.4.13}}{\leq} e^{O(T^2\varepsilon^2/d + \sqrt{Td\varepsilon} - d - T\varepsilon)}.$$

The existence of an (ε, δ)-learner implies $P^{opt}(\mathcal{E}) \geq 1 - \delta$. Since $P^{opt}(\mathcal{E})^2 \leq P^{pgm}(\mathcal{E})$, the above quantity is $\Omega(1)$, which implies $T \geq \Omega(d/\varepsilon)$. □

4.3 Sample complexity of agnostic learning

The following theorem characterizes the classical sample complexity of agnostic learning in terms of the VC dimension.

Theorem 4.14 ([VC74, Sim96, Tal94])**.** *Let $\mathcal{C}$ be a concept class with VC-dim$(\mathcal{C}) = d$. Then, $\Theta\Big(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\Big)$ examples are necessary and sufficient for an (ε, δ)-agnostic learner for $\mathcal{C}$.*

The lower bound was proven by Vapnik and Chervonenkis [VC74] (see also Simon [Sim96]), and the upper bound was proven by Talagrand [Tal94]. Shalev-Shwartz and Ben-David [SB14, Section 6.4] call Theorems 4.11 and 4.14 the “Fundamental Theorem of PAC learning”.

It turns out that the quantum sample complexity of agnostic learning is equal (up to constant factors) to the classical sample complexity. The proof of the lower bound is similar to the proof of the PAC case.

Theorem 4.15 ([AW16])**.** *Let $\mathcal{C}$ be a concept class with VC-dim$(\mathcal{C}) = d$. Then, for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/10)$, $\Omega\Big(\frac{d}{\varepsilon^2} + \frac{1}{\varepsilon^2}\log\frac{1}{\delta}\Big)$ examples are necessary for an (ε, δ)-quantum agnostic learner for $\mathcal{C}$.*

Proof sketch. We omit the easy proof of the d-independent part in the lower bound. In order to prove the $\Omega(d/\varepsilon^2)$ part, similar to the proof of Theorem 4.12, consider a $[d,k,r]_2$ linear code (for $k \geq d/4$, $r \geq d/8$) with generator matrix $M \in \mathbb{F}_2^{d \times k}$. Let $\{Mz : z \in \{0,1\}^k\}$ be the set of 2^k codewords, these have Hamming distance $d_H(Mz, My) \geq d/8$ whenever $z \neq y$. To each $z \in \{0,1\}^k$ we associate a distribution D_z:

$$D_z(s_i, b) = \frac{1}{d}\Big(\frac{1}{2} + 10(-1)^{(Mz)_i + b}\varepsilon\Big), \qquad \text{for } (i,b) \in [d] \times \{0,1\},$$

where $\mathcal{S} = \{s_1, \ldots, s_d\}$ is shattered by $\mathcal{C}$. Let $c^z \in \mathcal{C}$ be a concept that labels $\mathcal{S}$ according to $Mz \in \{0,1\}^d$. It is easy to see that c^z is the minimal-error concept in $\mathcal{C}$ w.r.t. the distribution D_z. Also, any learner that labels $\mathcal{S}$ according to $\ell \in \{0,1\}^d$ has an additional error $d_H(Mz, \ell) \cdot 20\varepsilon/d$

compared to c^z. Hence, with probability at least $1-\delta$, an (ε,δ)-quantum agnostic learner will find a labeling ℓ such that $d_H(Mz,\ell) \leq d/20$. Like in the proof of Theorem 4.12, because Mz is a codeword, finding an ℓ satisfying $d_H(Mz,\ell) \leq d/20$ is equivalent to *identifying* Mz (and hence z).

Now consider the following state identification problem: let $|\psi_z\rangle = \sum_{(i,b)\in[d]\times\{0,1\}} \sqrt{D_z(s_i,b)}|s_i,b\rangle$ for $z \in \{0,1\}^k$ and $\mathcal{E} = \{(2^{-k}, |\psi_z\rangle^{\otimes T})\}_{z\in\{0,1\}^k}$. Let G be the Gram matrix for this $\mathcal{E}$. We have

$$G(z,y) = \langle\psi_z|\psi_y\rangle^T = \frac{1}{2^k}\Big(1 - \frac{1-\sqrt{1-100\varepsilon^2}}{d}|M(z\oplus y)|\Big)^T.$$

Hence, the (z,y)-entry of G depends only on $z \oplus y$ and we are in a position to use Theorem 4.13. Similar to the proof of Theorem 4.12, we obtain

$$P^{pgm}(\mathcal{E}) = \sum_{z\in\{0,1\}^k} \sqrt{G}(z,z)^2 \overset{\text{Thm.4.13}}{\leq} e^{O(T^2\varepsilon^4/d+\sqrt{Td\varepsilon^2}-d-T\varepsilon^2)}.$$

This then implies $T = \Omega(d/\varepsilon^2)$ and proves the theorem. □

We just saw that in sample complexity for the PAC and agnostic models, quantum examples do not provide an advantage. Gavinsky [Gav12] introduced a model of learning called "Predictive Quantum" (PQ), a variation of the quantum PAC model. He exhibited a *relational* concept class that is polynomial-time learnable in PQ, while any "reasonable" classical model requires an exponential number of labeled examples to learn the class.

4.4 The learnability of quantum states

In addition to learning *classical* objects such as Boolean functions, one may also consider the learnability of *quantum* objects. Aaronson [Aar07] studied how well a quantum state ρ can be learned from measurement results. We are assuming here that each measurement is applied to ρ itself, so we require as many fresh copies of ρ as the number of measurements used. The goal is to end up with a *classical description* of a quantum state σ that is in some sense *close* to ρ—and which sense of "closeness" we require makes a huge difference. Learning such a good approximation of ρ in trace distance is called *state tomography.*

In general, an n-qubit state ρ is a Hermitian $2^n \times 2^n$ matrix of trace 1, and hence described by roughly 2^{2n} real parameters. For simplicity, let us restrict attention to allowing only two-outcome measurements on the state (Aaronson discusses also the more general case). Such a measurement is specified by two positive semi-definite operators E and $\mathrm{Id} - E$, and the probability for the measurement to yield the first outcome is $\mathrm{Tr}(E\rho)$. Since a two-outcome measurement gives at most one bit of information about ρ, $\Omega(2^{2n})$ measurement results are necessary to learn a σ that is very close to ρ in trace distance or Frobenius norm. Recently it was shown that such a number of copies is also *sufficient* [OW16, HHJ$^+$16].

Because of the exponential scaling in the number of qubits, the number of measurements needed for tomography of an arbitrary state on, say, 100 qubits is already prohibitively large. However, Aaronson showed an interesting and surprisingly efficient PAC-like result: from $O(n)$ measurement results, with measurements chosen i.i.d. according to an unknown distribution D on the set of all possible two-outcome measurements, we can construct an n-qubit quantum state σ that has roughly the same expectation value as ρ for "most" two-outcome measurements. In the latter, "most" is again measured under the same D that generated the measurements, just like in the usual PAC setting where the "approximate correctness" of the learner's hypothesis is evaluated under the same distribution D that generated the learner's examples. The output state σ can then be used to predict the behavior of ρ on two-outcome measurements, and it will give a good prediction for most measurements. Accordingly, $O(n)$ rather than $\exp(n)$ measurement results suffice for "pretty good tomography": to approximately learn an n-qubit state that is, maybe not close to ρ in trace distance, but still good enough for most practical purposes. More precisely, Aaronson's result is the following.

Theorem 4.16 ([Aar07]). *For every $\delta, \varepsilon, \gamma > 0$ there exists a learner with the following property: for every distribution D on the set of two-outcome measurements, given $T = n \cdot \mathrm{poly}(1/\varepsilon, 1/\gamma, \log(1/\delta))$ measurement results $(E_1, b_1), \ldots, (E_T, b_T)$ where each E_i is drawn i.i.d. from D and b_i is a bit with $\Pr[b_i = 1] = \mathrm{Tr}(E_i\rho)$, with probability $\geq 1 - \delta$ the learner produces the classical description of a state σ such that*

$$\Pr_{E \sim D}[|\mathrm{Tr}(E\sigma) - \mathrm{Tr}(E\rho)| > \gamma] \leq \varepsilon.$$

Note that the "approximately correct" motivation of the original PAC model is now quantified by two parameters ε and γ, rather than only by one parameter ε as before: the output state σ is deemed approximately correct if the value $\mathrm{Tr}(E\sigma)$ has additive error at most γ (compared to the correct value $\mathrm{Tr}(E\rho)$), except with probability ε over the choice of E. We then want the output to be approximately correct except with probability δ, like before. Note also that the theorem only says anything about the *sample* complexity of the learner (i.e., the number T of measurement results used to construct σ), not about the time complexity, which may be quite bad in general.

Proof sketch. The proof invokes general results due to Anthony and Bartlett [AB00] and Bartlett and Long [BL98] about learning classes of probabilistic functions[11] in terms of their *γ-fat-shattering dimension*. This generalizes VC dimension from Boolean to real-valued functions, as follows. For some set $\mathcal{E}$, let $\mathcal{C}$ be a class of functions $f : \mathcal{E} \to [0, 1]$. We say that the set $S = \{E_1, \ldots, E_d\} \subseteq \mathcal{E}$ is *γ-fat-shattered* by $\mathcal{C}$ if there exist $\alpha_1, \ldots, \alpha_d \in [0, 1]$ such that for all $Z \subseteq [d]$ there is an $f \in \mathcal{C}$ satisfying:

1. If $i \in Z$, then $f(E_i) \geq \alpha_i + \gamma$.
2. If $i \notin Z$, then $f(E_i) \leq \alpha_i - \gamma$.

The *γ-fat-shattering* dimension of $\mathcal{C}$ is the size of a largest S that is shattered by $\mathcal{C}$.[12]

For the application to learning quantum states, let $\mathcal{E}$ be the set of all n-qubit measurement operators. The relevant class of probabilistic functions corresponds to the n-qubit density matrices:

$$\mathcal{C} = \{f : \mathcal{E} \to [0, 1] \mid \exists\ n\text{-qubit } \rho \text{ s.t. } \forall E \in \mathcal{E}, f(E) = \mathrm{Tr}(E\rho)\}.$$

Suppose the set $S = \{E_1, \ldots, E_d\}$ is γ-fat-shattered by $\mathcal{C}$. This means that for each string $z \in \{0, 1\}^d$, there exists an n-qubit state ρ_z from which the bit z_i can be recovered using measurement E_i, with a γ-advantage over just outputting 1 with probability α_i. Such encodings $z \mapsto \rho_z$ of classical strings into quantum states are called *quantum random access codes.* Using known bounds on such codes [ANTV02], Aaronson shows that $d = O(n/\gamma^2)$. This upper bound on

[11] A probabilistic function f over a set $\mathcal{S}$ is a function $f : \mathcal{S} \to [0, 1]$.

[12] Note that if the functions in $\mathcal{C}$ have range $\{0, 1\}$ and $\gamma > 0$, then this is just our usual VC dimension.

the γ-fat-shattering dimension of $\mathcal{C}$ can then be plugged into [AB00, BL98] to get the theorem. □

More recently, in a similar spirit of learning quantum objects, Cheng et al. [CHY16] studied how many states are sufficient to learn an unknown *quantum measurement*. Here the answer turns out to be linear in the *dimension* of the space, so exponential in the number of qubits. However, learning an unknown quantum state becomes a *dual problem* to their question and using this connection they can reprove the results of Aaronson [Aar07] in a different way.

5 Time complexity

In many ways, the best measure of efficient learning is low *time complexity*. While low sample complexity is a necessary condition for efficient learning, the information-theoretic sufficiency of a small sample is not much help in practice if *finding* a good hypothesis still takes much time.[13] In this section we describe a number of results where the best quantum learner has much lower time complexity than the best known classical learner.

5.1 Time-efficient quantum PAC learning

When trying to find examples of quantum speed-ups for learning, it makes sense to start with the most famous example of quantum speed-up we have: Shor's algorithm for factoring integers in polynomial time [Sho97]. It is widely assumed that classical computers cannot efficiently factor Blum integers (i.e., integers that are the product of two distinct primes of equal bit-length, each congruent to 3 mod 4).

Prior to Shor's discovery, Kearns and Valiant [KV94a] had already constructed a concept class $\mathcal{C}$ based on factoring, as an example of a simple and efficiently-representable concept class with small VC dimension that is not efficiently learnable. Roughly speaking, each concept $c \in \mathcal{C}$ corresponds to a Blum integer N, and a positively-labeled example for the concept reveals N. A concise description of c, however, depends on the factorization of N, which is assumed to be hard to compute by classical computers. Servedio and Gortler [SG04] observed that, thanks to Shor's algorithm, this class *is* efficiently PAC learnable by quantum computers. They similarly observed that the factoring-based concept class devised by Angluin and Kharitonov [AK95] to show hardness of learning even with membership queries, *is* easy to learn by quantum computers.

Theorem 5.1 ([SG04]). *If there is no efficient classical algorithm for factoring Blum integers, then*

1. *there exists a concept class that is efficiently PAC learnable by quantum computers but not by classical computers;*

2. *there exists a concept class that is efficiently exactly learnable from membership queries by quantum computers but not by classical computers.*

One can construct classical one-way functions based on the assumption that factoring is hard. These functions can be broken (i.e., efficiently inverted) using quantum computers. However, there are other classical one-way functions that we do not known how to break with a quantum

[13] As is often the case: for many concept classes, finding a polynomial-sized hypothesis h that is consistent with a given set of examples is NP-hard.

computer. Surprisingly, Servedio and Gortler [SG04] managed to construct concept classes with quantum-classical separation based on any classical one-way function—irrespective of whether that one-way function can be broken by a quantum computer! The construction builds concepts by combining instances of Simon's problem [Sim97] with the pseudorandom function family that one can obtain from the one-way function.

Theorem 5.2 ([SG04]). *If classical one-way functions exist, then there is a concept class $\mathcal{C}$ that is efficiently exactly learnable from membership queries by quantum computers but not by classical computers.*

5.2 Learning DNF from uniform quantum examples

As we saw in Section 3, Bshouty and Jackson [BJ99] introduced the model of learning from quantum examples. Their main positive result is to show that Disjunctive Normal Form (DNF) formulas are learnable in polynomial time from quantum examples under the uniform distribution. For learning DNF under the uniform distribution from *classical* examples, the best upper bound is quasi-polynomial time [Ver90]. With the added power of *membership queries*, where the learner can actively ask for the label of any x of his choice, DNF formulas are known to be learnable in polynomial time under uniform D [Jac97], but polynomial-time learnability *without* membership queries is a longstanding open problem.

The classical polynomial-time algorithm for learning DNF using membership queries is Jackson's *harmonic sieve* algorithm [Jac97]. Roughly speaking it does the following. First, one can show that if the target concept $c : \{0,1\}^n \to \{0,1\}$ is an s-term DNF (i.e., a disjunction of at most s conjunctions of variables and negated variables) then there exists an n-bit parity function that agrees with c on a $1/2 + \Omega(1/s)$ fraction of the 2^n inputs. Moreover, the Goldreich-Levin algorithm [GL89] can be used to efficiently *find* such a parity function with the help of membership queries. This constitutes a "weak learner": an algorithm to find a hypothesis that agrees with the target concept with probability at least $1/2 + 1/\mathrm{poly}(s)$. Second, there are general techniques known as "boosting" [Fre95] that can convert a weak learner into a "strong" learner, i.e., one that produces a hypothesis that agrees with the target with probability $1 - \varepsilon$ rather than probability $1/2 + 1/\mathrm{poly}(s)$. Typically such boosting algorithms assume access to a weak learner that can produce a weak hypothesis under every possible distribution D, rather than just uniform D. The idea is to start with distribution $D_1 = D$, and use the weak learner to learn a weak hypothesis h_1 w.r.t. D_1. Then define a new distribution D_2 focusing on the inputs where the earlier hypothesis failed; use the weak learner to produce a weak hypothesis h_2 w.r.t. D_2, and so on. After $r = \mathrm{poly}(s)$ such steps the overall hypothesis h is defined as a majority function applied to $(h_1, \ldots, h_r)$.[14] Note that when learning under fixed uniform D, we can only sample the first distribution $D_1 = D$ directly. Fortunately, if one looks at the subsequent distributions $D_2, D_3, \ldots, D_r$ produced by boosting in this particular case, sampling those distributions D_i can be efficiently "simulated" using samples from the uniform distribution. Putting these ideas together yields a classical polynomial-time learner for DNF under the uniform distribution, using membership queries.

The part of the classical harmonic sieve that uses membership queries is the Goldreich-Levin algorithm for finding a parity (i.e., a character function χ_S) that is a weak hypothesis. The key to the *quantum* learner is to observe that one can replace Goldreich-Levin by Fourier sampling from uniform quantum examples (see Section 2.2.2). Let $f = 1-2c$, which is just c in ± 1-notation. If χ_S

[14] Note that this is not *proper* learning: the hypothesis h need not be an s-term DNF itself.

has correlation $\Omega(1/s)$ with the target, then $\widehat{f}(S) = \Omega(1/s)$ and Fourier sampling outputs that S with probability $\Omega(1/s^2)$. Hence poly(s) runs of Fourier sampling will with high probability give us a weak hypothesis. Because the state at step 3 of the Fourier sampling algorithm can be obtained with probability 1/2 from a uniform quantum example, we do not require the use of membership queries anymore. Describing this algorithm (and the underlying classical harmonic sieve) in full detail is beyond the scope of this survey, but the above sketch hopefully gives the main ideas of the result of [BJ99].

Theorem 5.3 ([BJ99])**.** *The concept class of s-term DNF is efficiently PAC learnable under the uniform distribution from quantum examples.*

5.3 Learning linear functions and juntas from uniform quantum examples

Uniform quantum examples can be used for learning other things as well. For example, suppose $f(x) = a \cdot x \bmod 2$ is a linear function over $\mathbb{F}_2$. Then the Fourier spectrum of f, viewed as a ± 1-valued function, has all its weight on χ_a. Hence by Fourier sampling we can perfectly recover a with $O(1)$ quantum sample complexity and $O(n)$ time complexity. In contrast, classical learners need $\Omega(n)$ examples to learn f, for the simple reason that each classical example (and even each membership query, if those are available to the learner too) gives at most one bit of information about the target concept.

A more complicated and interesting example is learning functions that depend (possibly nonlinearly) on at most k of the n input bits, with $k \ll n$. Such functions are called *k-juntas*, since they are "governed" by a small subset of the input bits. We want to learn such f up to error ε from uniform (quantum or classical) examples. A trivial learner would sample $O(2^k \log n)$ classical examples and then go over all $\binom{n}{k}$ possible sets of up k variables in order to find one that is consistent with the sample. This gives time complexity $O(n^k)$. The best known upper bound on time complexity [MOS04] is only slightly better: $O(n^{k\omega/(\omega+1)})$, where $\omega \in [2, 2.38]$ is the optimal exponent for matrix multiplication.

Time-efficiently learning k-juntas under the uniform distribution for $k = O(\log n)$ is a notorious bottleneck in classical learning theory, since it is a special case of DNF learning: every k-junta can be written as an s-term DNF with $s < 2^k$, by just taking the OR over the 1-inputs of the underlying k-bit function. In particular, if we want to efficiently learn poly(n)-term DNF from uniform examples (still an open problem, as mentioned in the previous section) then we should at least be able to efficiently learn $O(\log n)$-juntas (also still open).

Bshouty and Jackson's DNF learner from uniform quantum examples implies that we can learn k-juntas using poly$(2^k, n)$ quantum examples and time (for fixed ε, δ). Atıcı and Servedio [AS09] gave a more precise upper bound.

Theorem 5.4 ([AS09])**.** *There exists a quantum algorithm for learning k-juntas under the uniform distribution that uses $O(k \log(k)/\varepsilon)$ uniform quantum examples, $O(2^k)$ uniform classical examples, and $O(nk \log(k)/\varepsilon + 2^k \log(1/\varepsilon))$ time.*

Proof sketch. The idea is to first use Fourier sampling from quantum examples to find the k variables (at least the ones with non-negligible influence), and then to use $O(2^k)$ uniform classical examples to learn (almost all of) the truth-table of the function on those variables.

View the target k-junta f as a function with range ±1. Let the *influence* of variable x_i on f be

$$\mathrm{Inf}_i(f) = \sum_{S:S_i=1} \widehat{f}(S)^2 = \mathbb{E}_x\Big[\Big(\frac{f(x)-f(x\oplus e_i)}{2}\Big)^2\Big] = \Pr_x[f(x)\neq f(x\oplus e_i)],$$

where $x \oplus e_i$ is x after flipping its ith bit. If $S_i = 1$ for an i that is not in the junta, then $\widehat{f}(S) = 0$. Hence Fourier sampling returns an S such that $S_i = 1$ only for variables in the junta. $\mathrm{Inf}_i(f)$ is exactly the probability that $S_i = 1$. Hence for a fixed i, the probability that i does *not* appear in T Fourier samples is

$$(1-\mathrm{Inf}_i(f))^T \leq e^{-T\,\mathrm{Inf}_i(f)}.$$

If we set $T = O(k\log(k)/\varepsilon)$ and let V be the union of the supports of the T Fourier samples, then with high probability V contains all junta variables except those with $\mathrm{Inf}_i(f) \ll \varepsilon/k$ (the latter ones can be ignored since even their joint influence is negligible).

Now use $O(2^k\log(1/\varepsilon))$ uniform classical examples. With high probability, at least $1-\varepsilon/2$ of all $2^{|V|}$ possible settings of the variables in V will appear, and we use those to formulate our hypothesis h (say with random values for the few inputs of the truth-table that we didn't see in our sample, and for the ones that appeared twice with inconsistent f-values). One can show that, with high probability, h will disagree with f on at most an ε-fraction of $\{0,1\}^n$. □

In a related result, Belovs [Bel15] gives a very tight analysis of the number of quantum membership queries (though not the time complexity) needed to exactly learn k-juntas whose underlying k-bit function is symmetric. For example, if the k-bit function is OR or Majority, then $O(k^{1/4})$ quantum membership queries suffice. For the case of Majority, $\Theta(k)$ classical membership queries are required, giving a fourth-power separation between quantum and classical membership query complexity of exact learning (see Corollary 4.4).

6 Conclusion

Quantum learning theory studies the theoretical aspects of quantum machine learning. We surveyed what is known about this area. Specifically

- **Query complexity of exact learning.** The number of quantum membership queries needed to exactly learn a target concept can be polynomially smaller than the number of classical membership queries, but not much smaller than that.

- **Sample complexity.** For the distribution-independent models of PAC and agnostic learning, quantum examples give no significant advantage over classical random examples: for every concept class, the classical and quantum sample complexities are the same up to constant factors. In contrast, for some fixed distributions (e.g., uniform) quantum examples can be much better than classical examples.

- **Time complexity.** There exist concept classes that can be learned superpolynomially faster by quantum computers than by classical computers, for instance based on Shor's or Simon's algorithm. This holds both in the model of exact learning with membership queries, and in the model of PAC-learning. If one allows uniform quantum examples, DNF and juntas can be learned much more efficiently than we know how to do classically.

We end with a number of directions for future research.

- Bshouty and Jackson [BJ99] showed that DNF (i.e., disjunctions of conjunctions of variables and negations of variables) can be efficiently learned from uniform quantum examples. Is the same true of depth-3 circuits? And what about *constant-depth* circuits with unbounded fan-in AND/OR or even threshold gates, i.e., the concept classes AC^0 and TC^0—might even these be efficiently learnable from uniform quantum examples or even PAC-learnable? The latter is one of Scott Aaronson's "Ten Semi-Grand Challenges for Quantum Computing Theory" [Aar05]. Classically, the best upper bounds on time complexity of learning AC^0 are quasi-polynomial under the uniform distribution [LMN93], and roughly $\exp(n^{1/3})$ in the PAC model (i.e., under all possible distributions) [KS04]; see [DS16] for a recent hardness result.

- Atıcı and Servedio [AS05] asked if for every $\mathcal{C}$, the upper bound in Corollary 4.4 can be improved to $D(\mathcal{C}) \leq O(nQ(\mathcal{C}) + Q(\mathcal{C})^2)$?

- Can we characterize the classical and quantum query complexity of exactly learning a concept class $\mathcal{C}$ in terms of the combinatorial parameter $\gamma(\mathcal{C})$, or in terms of the extended teaching dimension of $\mathcal{C}$?

- Can we find more instances of concept classes where quantum examples are beneficial when learning w.r.t. some fixed distribution (uniform or otherwise), or some restricted set of distributions?

- Can we find examples of quantum speed-up in Angluin's [Ang87] model of equivalence queries plus membership queries?

- Most research in quantum learning theory (and hence this survey) has focused on concept classes of Boolean functions. What about learning classes of *real-valued* or even *vector-valued* functions?

- Can we find a *proper* quantum PAC learner with optimal sample complexity, i.e., one whose output hypothesis lies in $\mathcal{C}$ itself? Or a *proper* efficient quantum learner for DNF using uniform quantum examples?

- Can we find practical machine learning problems with a large provable quantum speed-up?

- Can we use quantum machine learning for "quantum supremacy", i.e., for solving some task using 50–100 qubits in a way that is convincingly faster than possible on large classical computers? (See for example [AC16] for some complexity results concerning quantum supremacy.)

Acknowledgments.

We thank Lane Hemaspaandra for commissioning this survey for the SIGACT News Complexity Theory Column and helpful comments, and Robin Kothari for helpful comments and pointers (including to [Aar05]), and for sending us his 2012 manuscript [Kot12].

References

[AAD$^+$15] J. Adcock, E. Allen, M. Day, S. Frick, J. Hinchliff, M. Johnson, S. Morley-Short, S. Pallister, A. Price, and S. Stanisic. Advances in quantum machine learning, 9 Dec 2015. arXiv:1512.02900.

[Aar05] S. Aaronson. Ten semi-grand challenges for quantum computing theory. `http://www.scottaaronson.com/writings/qchallenge.html`, 2005.

[Aar07] S. Aaronson. The learnability of quantum states. *Proceedings of the Royal Society of London*, 463(2088), 2007. quant-ph/0608142.

[Aar15] S. Aaronson. Quantum machine learning algorithms: Read the fine print. *Nature Physics*, 11(4):291–293, April 2015.

[AB00] M. Anthony and P. Bartlett. Function learning from interpolation. *Combinatorics, Probability, and Computing*, 9(3):213–225, 2000. Earlier version in EuroCOLT'95.

[AB09] M. Anthony and P. L. Bartlett. *Neural network learning: Theoretical foundations.* Cambridge University Press, 2009.

[ABG06] E. Aïmeur, G. Brassard, and S. Gambs. Machine learning in a quantum world. In *Proceedings of Advances in Artificial Intelligence, 19th Conference of the Canadian Society for Computational Studies of Intelligence*, volume 4013 of *Lecture Notes in Artificial Intelligence*, pages 431–442, 2006.

[ABG13] E. Aïmeur, G. Brassard, and S. Gambs. Quantum speed-up for unsupervised learning. *Machine Learning*, 90(2):261–287, 2013.

[AC16] S. Aaronson and L. Chen. Complexity-theoretic foundations of quantum supremacy experiments. arXiv:1612.05903, 2016.

[AIK$^+$04] A. Ambainis, K. Iwama, A. Kawachi, H. Masuda, R. H. Putra, and S. Yamashita. Quantum identification of Boolean oracles. In *Proceedings of 30th Annual Symposium on Theoretical Aspects of Computer Science (STACS'04)*, pages 105–116, 2004. arXiv:quant-ph/0403056.

[AIK$^+$07] A. Ambainis, K. Iwama, A. Kawachi, R. Raymond, and S. Yamashita. Improved algorithms for quantum identification of Boolean oracles. *Theoretical Computer Science*, 378(1):41–53, 2007.

[AIN$^+$09] A. Ambainis, K. Iwama, M. Nakanishi, H. Nishimura, R. Raymond, S. Tani, and S. Yamashita. Average/worst-case gap of quantum query complexities by on-set size. 2009. arXiv:0908.2468v1.

[AK95] D. Angluin and M. Kharitonov. When won't membership queries help? *Journal of Computer and System Sciences*, 50(2):336–355, 1995. Earlier version in STOC'91.

[Amb02] A. Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002. Earlier version in STOC'00. quant-ph/0002066.

[Ang87] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1987.

[ANTV02] A. Ambainis, A. Nayak, A. Ta-Shma, and U. V. Vazirani. Dense quantum coding and quantum finite automata. *Journal of the ACM*, 49(4):496–511, 2002. Earlier version in STOC'99.

[AS05] A. Atıcı and R. Servedio. Improved bounds on quantum learning algorithms. *Quantum Information Processing*, 4(5):355–386, 2005. quant-ph/0411140.

[AS09] A. Atıcı and R. Servedio. Quantum algorithms for learning and testing juntas. *Quantum Information Processing*, 6(5):323–348, 2009. arXiv:0707.3479.

[AW16] S. Arunachalam and R. de Wolf. Optimal quantum sample complexity of learning algorithms, 4 Jul 2016. arXiv:1607.00932.

[BBBV97] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. quant-ph/9701001.

[BCG$^+$96] N. H. Bshouty, R. Cleve, R. Gavaldà, S. Kannan, and C. Tamon. Oracles and queries that are sufficient for exact learning. *Journal of Computer and System Sciences*, 52(3):421–433, 1996. Earlier version in COLT'94.

[BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[Bel15] A. Belovs. Quantum algorithms for learning symmetric juntas via the adversary bound. *Computational Complexity*, 24(2):255–293, 2015. Earlier version in Complexity'14. arXiv:1311.6777.

[Ben82] P. A. Benioff. Quantum mechanical Hamiltonian models of Turing machines. *Journal of Statistical Physics*, 29(3):515–546, 1982.

[BHMT02] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, pages 53–74. 2002. quant-ph/0005055.

[BJ99] N. H. Bshouty and J. C. Jackson. Learning DNF over the uniform distribution using a quantum example oracle. *SIAM Journal on Computing*, 28(3):1136–1153, 1999. Earlier version in COLT'95.

[BK02] H. Barnum and E. Knill. Reversing quantum dynamics with near-optimal quantum and classical fidelity. *Journal of Mathematical Physics*, 43:2097–2106, 2002. quant-ph/0004088.

[BL98] P. Bartlett and P. M. Long. Prediction, learning, uniform convergence, and scale-sensitive dimensions. *Journal of Computer and System Sciences*, 56(2):174–190, 1998.

[BV97] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. Earlier version in STOC'93.

[BWP$^+$16] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. Quantum machine learning, 28 Nov 2016. arXiv:1611.09347.

[CHY16] H. C. Cheng, M. H. Hsieh, and P. C. Yeh. The learnability of unknown quantum measurements. *Quantum Information and Computation*, 16(7&8):615–656, 2016. arXiv:1501.00559.

[Deu85] D. Deutsch. Quantum theory, the Church-Turing principle, and the universal quantum Turing machine. In *Proceedings of the Royal Society of London*, volume A400, pages 97–117, 1985.

[DS16] A. Daniely and S. Shalev-Shwartz. Complexity theoretic limitations on learning DNF's. In *Proceedings of the 29th Conference on Learning Theory (COLT'16)*, 2016.

[Fey82] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467–488, 1982.

[Fey85] R. Feynman. Quantum mechanical computers. *Optics News*, 11:11–20, 1985.

[Fre95] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995. Earlier version in COLT'90.

[Gav12] D. Gavinsky. Quantum predictive learning and communication complexity with single input. *Quantum Information and Computation*, 12(7-8):575–588, 2012. Earlier version in COLT'10. arXiv:0812.3429.

[GL89] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of 21st ACM STOC*, pages 25–32, 1989.

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th ACM STOC*, pages 212–219, 1996. quant-ph/9605043.

[Han16] S. Hanneke. The optimal sample complexity of PAC learning. *Journal of Machine Learning Research*, 17(38):1–15, 2016. arXiv:1507.00473.

[Hau92] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

[Heg95] T. Hegedűs. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the 8th Conference on Learning Theory (COLT'95)*, pages 108–117, 1995.

[HHJ$^+$16] J. Haah, A. W. Harrow, Z. Ji, X. Wu, and N. Yi. Sample-optimal tomography of quantum states. In *Proceedings of 48th ACM STOC*, pages 913–925, 2016. arXiv:1508.01797.

[HHL09] A. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for solving linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. arXiv:0811.3171.

[HMP$^+$10] M. Hunziker, D. A. Meyer, J. Park, J. Pommersheim, and M. Rothstein. The geometry of quantum learning. *Quantum Information Processing*, 9(3):321–341, 2010. quant-ph/0309059.

[Hol73] A. S. Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problemy Peredachi Informatsii*, 9(3):3–11, 1973. English translation in *Problems of Information Transmission*, 9:177–183, 1973.

[Jac97] J. C. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997. Earlier version in FOCS'94.

[Kot12] R. Kothari. Quantum computing and learning theory. Unpublished manuscript, 2012.

[Kot14] R. Kothari. An optimal quantum algorithm for the oracle identification problem. In *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*, pages 482–493, 2014. arXiv:1311.7685.

[KP17] I. Kerenidis and A. Prakash. Quantum recommendation systems. In *Innovations in Theoretical Computer Science (ITCS'17)*, 2017. arXiv:1603.08675.

[KS04] A. Klivans and R. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. *Journal of Computer and System Sciences*, 68(2):303–318, 2004. Earlier version in STOC'01.

[KSS94] M. J. Kearns, R. E. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994. Earlier version in COLT'92.

[KV94a] M. J. Kearns and L. G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

[KV94b] M. J. Kearns and U. V. Vazirani. *An introduction to computational learning theory.* MIT Press, 1994.

[LL16] C. Y.-Y. Lin and H. Lin. Upper bounds on quantum query complexity inspired by the Elitzur-Vaidman bomb tester. *Theory of Computing*, 12:1–35, 2016. Earlier version in CCC'15. arXiv:1410.0932.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM*, 40(3):607–620, 1993. Earlier version in FOCS'89.

[LMR13a] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum algorithms for supervised and unsupervised machine learning, 1 Jul 2013. arXiv:1307.0411.

[LMR13b] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(631–633), 2013. arXiv:1307.0401.

[Man80] Y. Manin. Vychislimoe i nevychislimoe (computable and noncomputable). *Soviet Radio*, pages 13–15, 1980. In Russian.

[Man99] Y. Manin. Classical computing, quantum computing, and Shor's factoring algorithm. quant-ph/9903008, 2 Mar 1999.

[Mon07] A. Montanaro. On the distinguishability of random quantum states. *Communications in Mathematical Physics*, 273(3):619–636, 2007. quant-ph/0607011.

[Mos83] M. Yu. Moshkov. Conditional tests. *Problemy Kibernetzkt*, 40:131–170, 1983. In Russian.

[MOS04] E. Mossel, R. O'Donnell, and R. Servedio. Learning functions of k relevant variables. *Journal of Computer and System Sciences*, 69(3):421–434, 2004. Earlier version in STOC'03.

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[O'D14] R. O'Donnell. *Analysis of Boolean Functions.* Cambridge University Press, 2014.

[OW16] R. O'Donnell and J. Wright. Efficient quantum tomography. In *Proceedings of 48th ACM STOC*, pages 899–912, 2016. arXiv:1508.01907.

[RML13] P. Rebentrost, M. Mohseni, and S. Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(13), 2013. arXiv:1307.0471.

[SB14] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms.* Cambridge University Press, 2014.

[SG04] R. Servedio and S. Gortler. Equivalences and separations between quantum and classical learnability. *SIAM Journal on Computing*, 33(5):1067–1092, 2004. Combines earlier papers from ICALP'01 and CCC'01. quant-ph/0007036.

[Sho97] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. Earlier version in FOCS'94. quant-ph/9508027.

[Sim96] H. U. Simon. General bounds on the number of examples needed for learning probabilistic concepts. *Journal of Computer and System Sciences*, 52(2):239–254, 1996. Earlier version in COLT'93.

[Sim97] D. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. Earlier version in FOCS'94.

[Sim15] H. U. Simon. An almost optimal PAC algorithm. In *Proceedings of the 28th Conference on Learning Theory (COLT)*, pages 1552–1563, 2015.

[SSP15] M. Schuld, I. Sinayskiy, and F. Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015. arXiv:1409.3097.

[Tal94] M. Talagrand. Sharper bounds for Gaussian and empirical processes. *The Annals of Probability*, pages 28–76, 1994.

[Val84] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[VC71] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971. English translation of 1968 Russian paper in *Dokl. Akad. Nauk.* 181(4).

[VC74] V. Vapnik and A. Chervonenkis. *Theory of pattern recognition.* Nauka, USSR, 1974. In Russian.

[Ver90] K. A. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory (COLT'90)*, pages 314–326, 1990.

[Wit14] P. Wittek. *Quantum Machine Learning: What Quantum Computing Means to Data Mining.* Elsevier, 2014.

[WKS16a] N. Wiebe, A. Kapoor, and K. Svore. Quantum deep learning. *Quantum Information and Computation*, 16(7):541–587, 2016. arXiv:1412.3489.

[WKS16b] N. Wiebe, A. Kapoor, and K. M. Svore. Quantum perceptron models, 2016. arXiv:1602.04799.

[Wol08] R. de Wolf. A brief introduction to Fourier analysis on the Boolean cube. *Theory of Computing*, 2008. ToC Library, Graduate Surveys 1.

[Zha10] C. Zhang. An improved lower bound on query complexity for quantum PAC learning. *Information Processing Letters*, 111(1):40–45, 2010.