Quantum versus Classical Learnability

Rocco A. Servedio Steven J. Gortler
Harvard University
Division of Engineering and Applied Sciences
33 Oxford Street
Cambridge, MA
{rocco,sjg}@cs.harvard.edu

Abstract

Motivated by recent work on quantum black-box query complexity, we consider quantum versions of two well-studied models of learning Boolean functions: Angluin's model of exact learning from membership queries and Valiant's Probably Approximately Correct (PAC) model of learning from random examples. For each of these two learning models we establish a polynomial relationship between the number of quantum versus classical queries required for learning. Our results provide an interesting contrast to known results which show that testing black-box functions for various properties can require exponentially more classical queries than quantum queries. We also show that under a widely held computational hardness assumption there is a class of Boolean functions which is polynomial-time learnable in the quantum version but not the classical version of each learning model; thus while quantum and classical learning are equally powerful from an information theory perspective, they are different when viewed from a computational complexity perspective.

1. Introduction

1.1. Motivation

In recent years many researchers have investigated the power of quantum computers which can query a black-box oracle for an unknown function [1, 5, 6, 9, 14, 10, 11, 15, 17, 20, 21, 23, 32, 37]. The broad goal of research in this area is to understand the relationship between the number of quantum versus classical oracle queries which are required to answer various questions about the function computed by the oracle. For example, a well-known result due to Deutsch and Jozsa [17] shows that exponentially fewer queries are required in the quantum model in order to determine with certainty whether a black-box oracle computes a constant Boolean function or a function which is balanced between outputs 0 and 1. More recently, several researchers have studied the number of quantum oracle queries which are required to determine whether the function computed by a black-box oracle is identically zero [5, 6, 9, 15, 23, 37].

A natural question which arises in this framework is the following: what is the relationship between the number of quantum versus classical oracle queries which are required in order to *exactly identify* the function computed by a black-box oracle? Here the goal is not to determine whether a black-box function satisfies some particular property such as ever taking a nonzero value, but rather to precisely identify an unknown black-box function from some restricted class of possible functions. The classical version of this problem has been well studied in the computational learning theory literature [2, 12, 22, 24, 25] and is known as the problem of *exact learning from membership queries*. The question stated above can thus be rephrased as follows: what is the relationship between the number of quantum versus classical membership queries which are required for exact learning? We answer this question in this paper.

In addition to the model of exact learning from membership queries, we also consider a quantum version of Valiant's widely studied PAC learning model which was introduced by Bshouty and Jackson [13]. While a learning algorithm in the classical PAC model has access to labeled examples drawn from some fixed probability distribution, a learning algorithm in the quantum PAC model has access to some fixed quantum superposition of labeled examples. Bshouty and Jackson gave a polynomial-time algorithm for a particular learning problem in the quantum PAC model, but did not address the general relationship between the number of quantum versus classical examples which are required for PAC learning. We answer this question as well.

1.2. Our results

We show that in an information-theoretic sense, quantum and classical learning are equivalent up to polynomial factors: for both the model of exact learning from membership queries and the PAC model, there is no learning problem which can be solved using significantly fewer quantum examples than classical examples. More precisely, our first main theorem is the following:

Theorem 1 *Let $\mathcal{C}$ be any class of Boolean functions over $\{0,1\}^n$ and let D and Q be such that $\mathcal{C}$ is exact learnable from D classical membership queries or from Q quantum membership queries. Then $D = O(nQ^3)$.*

Our second main theorem is an analogous result for quantum versus classical PAC learnability:

Theorem 2 *Let $\mathcal{C}$ be any class of Boolean functions over $\{0,1\}^n$ and let D and Q be such that $\mathcal{C}$ is PAC learnable from D classical examples or from Q quantum examples. Then $D = O(nQ)$.*

Theorems 1 and 2 are information-theoretic rather than computational in nature; they show that for any learning problem, if there is a quantum learning algorithm which uses polynomially many examples then there must also exist a classical learning algorithm which uses polynomially many examples. However, Theorems 1 and 2 do not imply that every *polynomial time* quantum learning algorithm must have a polynomial time classical analogue. In fact, we show that a separation exists between efficient quantum learnability and efficient clasical learnability. Under a widely held computational hardness assumption for classical computation (the hardness of factoring Blum integers), we observe that for each of the two learning models considered in this paper there is a concept class which is polynomial-time learnable in the quantum version but not in the classical version of the model.

1.3. Previous Work

Our results draw on lower bound techniques from both quantum computation and computational learning theory [2, 5, 6, 8, 12, 24]. A detailed description of the relationship between our results and previous work on quantum versus classical black-box query complexity is given in Section 3.4.

In [19] Farhi *et al.* prove a lower bound on the number of functions which can be distinguished with k quantum queries. Ronald de Wolf has noted [18] that the main result of [19] yields an alternate proof of one of the two lower bounds which we give for exact learning from quantum membership queries (Theorem 10).

1.4. Organization

We define the exact learning model and the PAC learning model and describe the quantum computation framework in Section 2. We prove the relationship between quantum and classical exact learning from membership queries (Theorem 1) in Section 3, and we prove the relationship between quantum and classical PAC learning (Theorem 2) in Section 4. Finally, in Section 5 we observe that under a widely accepted computational hardness assumption for classical computation, in each of these two learning models there is a concept class which is quantum learnable in polynomial time but not classically learnable in polynomial time.

2. Preliminaries

A *concept* c over $\{0,1\}^n$ is a Boolean function over the domain $\{0,1\}^n$, or equivalently a concept can be viewed as a subset $\{x \in \{0,1\}^n : c(x) = 1\}$ of $\{0,1\}^n$. A *concept class* $\mathcal{C} = \cup_{n \geq 1} C_n$ is a collection of concepts, where $C_n = \{c \in \mathcal{C} : \ c \text{ is a concept over } \{0,1\}^n\}$. For example, C_n might be the family of all Boolean formulae over n variables which are of size at most n^2. We say that a pair $\langle x, c(x)\rangle$ is a *labeled example* of the concept c.

While many different learning models have been proposed, most models follow the same basic paradigm: a learning algorithm for a concept class $\mathcal{C}$ typically has access to (some kind of) an oracle which provides examples that are labeled according to a fixed but unknown target concept $c \in \mathcal{C}$, and the goal of the learning algorithm is to infer (in some sense) the target concept c. The two learning models which we discuss in this paper, the model of exact learning from membership queries and the PAC model, make this rough notion precise in different ways.

2.1. Classical Exact Learning from Membership Queries

The model of *exact learning from membership queries* was introduced by Angluin [2] and has since been widely studied [2, 12, 22, 24, 25]. In this model the learning algorithm has access to a *membership oracle* MQ_c where $c \in C_n$ is the unknown target concept. When given an input string $x \in \{0,1\}^n$, in one time step the oracle MQ_c returns the bit $c(x)$; such an invocation is known as a *membership query* since the oracle's answer tells whether or not $x \in c$ (viewing c as a subset of $\{0,1\}^n$). The goal of the learning algorithm is to construct a hypothesis $h : \{0,1\}^n \to \{0,1\}$ which is logically equivalent to c, i.e. $h(x) = c(x)$ for all $x \in \{0,1\}^n$. Formally, we say that an algorithm A is an *exact learning algorithm for $\mathcal{C}$ using membership queries* if for all $n \geq 1$, for all $c \in C_n$, if A is given n and access to MQ_c, then with probability at least $2/3$ algorithm

A outputs a Boolean circuit h such that $h(x) = c(x)$ for all $x \in \{0,1\}^n$. The *sample complexity* $T(n)$ of a learning algorithm A for $\mathcal{C}$ is the maximum number of calls to MQ_c which A ever makes for any $c \in C_n$.

2.2. Classical PAC Learning

The PAC (Probably Approximately Correct) model of concept learning was introduced by Valiant in [33] and has since been extensively studied [4, 27]. In this model the learning algorithm has access to an *example oracle* $EX(c, \mathcal{D})$ where $c \in C_n$ is the unknown target concept and $\mathcal{D}$ is an unknown distribution over $\{0,1\}^n$. The oracle $EX(c, \mathcal{D})$ takes no inputs; when invoked, in one time step it returns a labeled example $\langle x, c(x)\rangle$ where $x \in \{0,1\}^n$ is randomly selected according to the distribution $\mathcal{D}$. The goal of the learning algorithm is to generate a hypothesis $h : \{0,1\}^n \to \{0,1\}$ which is an *ϵ-approximator for c under $\mathcal{D}$*, i.e. a hypothesis h such that $\Pr_{x\in\mathcal{D}}[h(x) \neq c(x)] \leq \epsilon$. An algorithm A is a *PAC learning algorithm for $\mathcal{C}$* if the following condition holds: for all $n \geq 1$ and $0 < \epsilon, \delta < 1$, for all $c \in C_n$, for all distributions $\mathcal{D}$ over $\{0,1\}^n$, if A is given n, ϵ, δ and access to $EX(c, \mathcal{D})$, then with probability at least $1 - \delta$ algorithm A outputs a circuit h which is an ϵ-approximator for c under $\mathcal{D}$. The *sample complexity* $T(n, \epsilon, \delta)$ of a learning algorithm A for $\mathcal{C}$ is the maximum number of calls to $EX(c, \mathcal{D})$ which A ever makes for any concept $c \in C_n$ and any distribution $\mathcal{D}$ over $\{0,1\}^n$.

2.3. Quantum Computation

Detailed descriptions of the quantum computation model can be found in [7, 16, 28, 36]; here we outline only the basics using the terminology of *quantum networks* as presented in [5]. A quantum network $\mathcal{N}$ is a quantum circuit (over some standard basis augmented with one oracle gate) which acts on an m-bit quantum register; the computational basis states of this register are the 2^m binary strings of length m. A quantum network can be viewed as a sequence of unitary transformations

$$U_0, O_1, U_1, O_2, \ldots, U_{T-1}, O_T, U_T,$$

where each U_i is an arbitrary unitary transformation on m qubits and each O_i is a unitary transformation which corresponds to an oracle call.[1] Such a network is said to have *query complexity* T. At every stage in the execution of the network, the current state of the register can be represented as a superposition $\sum_{z\in\{0,1\}^m} \alpha_z|z\rangle$ where the α_z are complex numbers which satisfy $\sum_{z\in\{0,1\}^m} \|\alpha_z\|^2 = 1$. If this state is measured, then with probability $\|\alpha_z\|^2$ the string $z \in \{0,1\}^m$ is observed and the state collapses down to $|z\rangle$. After the final transformation U_T takes place, a measurement is performed on some subset of the bits in the register and the observed value (a classical bit string) is the output of the computation.

Several points deserve mention here. First, since the information which our quantum network uses for its computation comes from the oracle calls, we may stipulate that the initial state of the quantum register is $|0^m\rangle$. Second, as described above each U_i can be an arbitrarily complicated unitary transformation (as long as it does not contain any oracle calls) which may require a large quantum circuit to implement. This is of small concern since we are chiefly interested in query complexity and not circuit size. Third, as defined above our quantum networks can make only one measurement at the very end of the computation; this is an inessential restriction since any algorithm which uses intermediate measurements can be modified to an algorithm which makes only one final measurement. Finally, we have not specified just how the oracle calls O_i work; we address this point separately in Sections 3.1 and 4.1 for each type of oracle.

If $|\phi\rangle = \sum_z \alpha_z|z\rangle$ and $|\psi\rangle = \sum_z \beta_z|z\rangle$ are two superpositions of basis states, then the *Euclidean distance* betweeen $|\phi\rangle$ and $|\psi\rangle$ is $||\phi\rangle - |\psi\rangle| = (\sum_z |\alpha_z - \beta_z|^2)^{1/2}$. The *total variation distance* between two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ is defined to be $\sum_x |\mathcal{D}_1(x) - \mathcal{D}_2(x)|$. The following fact (Lemma 3.2.6 of [7]), which relates the Euclidean distance between two superpositions and the total variation distance between the distributions induced by measuring the two superpositions, will be useful:

Fact 3 *Let $|\phi\rangle$ and $|\psi\rangle$ be two unit-length superpositions which represent possible states of a quantum register. If the Euclidean distance $||\phi\rangle - |\psi\rangle|$ is at most ϵ, then performing the same observation on $|\phi\rangle$ and $|\psi\rangle$ induces distributions $\mathcal{D}_\phi$ and $\mathcal{D}_\psi$ which have total variation distance at most 4ϵ.*

3. Exact Learning from Quantum Membership Queries

3.1. Quantum Membership Queries

A *quantum membership oracle* QMQ_c is the natural quantum generalization of a classical membership oracle MQ_c: on input a superposition of query strings, the oracle QMQ_c generates the corresponding superposition of example labels. More formally, a QMQ_c gate maps the basis state $|x, b\rangle$ (where $x \in \{0,1\}^n$ and $b \in \{0,1\}$) to the state $|x, b \oplus c(x)\rangle$. If $\mathcal{N}$ is a quantum network which has QMQ_c gates as its oracle gates, then each O_i is the unitary transformation which maps $|x, b, y\rangle$ (where $x \in \{0,1\}^n$, $b \in \{0,1\}$

[1] Since there is only one kind of oracle gate, each O_i is the same transformation.

and $y \in \{0,1\}^{m-n-1}$) to $|x, b \oplus c(x), y\rangle$.[2] Our QMQ_c oracle is identical to the well-studied notion of a quantum black-box oracle for c [5, 6, 7, 9, 10, 11, 15, 17, 23, 37].

A *quantum exact learning algorithm for* $\mathcal{C}$ is a family of quantum networks $\mathcal{N}_1, \mathcal{N}_2, \ldots$, where each network $\mathcal{N}_n$ has a fixed architecture independent of the choice of $c \in C_n$, with the following property: for all $n \geq 1$, for all $c \in C_n$, if $\mathcal{N}_n$'s oracle gates are instantiated as QMQ_c gates, then with probability at least 2/3 the network $\mathcal{N}_n$ outputs a representation of a (classical) Boolean circuit $h : \{0,1\}^n \to \{0,1\}$ such that $h(x) = c(x)$ for all $x \in \{0,1\}^n$. The *quantum sample complexity* of a quantum exact learning algorithm for $\mathcal{C}$ is $T(n)$, where $T(n)$ is the query complexity of $\mathcal{N}_n$.

3.2. Lower Bounds on Classical and Quantum Exact Learning

Two different lower bounds are known for the number of classical membership queries which are required to exact learn any concept class. In this section we prove two analogous lower bounds on the number of quantum membership queries required to exact learn any concept class. Throughout this section for ease of notation we omit the subscript n and write C for C_n.

A Lower Bound Based on Similarity of Concepts. Consider a set of concepts which are all "similar" in the sense that for every input almost all concepts in the set agree. Known results in learning theory state that such a concept class must require a large number of membership queries for exact learning. More formally, let $C' \subseteq C$ be any subset of C. For $a \in \{0,1\}^n$ and $b \in \{0,1\}$ let $C'_{\langle a,b\rangle}$ denote the set of those concepts in C' which assign label b to example a, i.e. $C'_{\langle a,b\rangle} = \{c \in C' : c(a) = b\}$. Let $\gamma^{C'}_{\langle a,b\rangle} = |C'_{\langle a,b\rangle}|/|C'|$ be the fraction of such concepts in C', and let $\gamma^{C'}_a = \min\{\gamma^{C'}_{\langle a,0\rangle}, \gamma^{C'}_{\langle a,1\rangle}\}$; thus $\gamma^{C'}_a$ is the minimum fraction of concepts in C' which can be eliminated by querying MQ_c on the string a. Let $\gamma^{C'} = \max\{\gamma^{C'}_a : a \in \{0,1\}^n\}$. Finally, let $\hat{\gamma}^C$ be the minimum of $\gamma^{C'}$ across all $C' \subseteq C$ such that $|C'| \geq 2$. Thus

$$\hat{\gamma}^C = \min_{C' \subseteq C, |C'| \geq 2} \; \max_{a \in \{0,1\}^n} \; \min_{b \in \{0,1\}} \frac{|C'{}_{\langle a,b\rangle}|}{|C'|}.$$

Intuitively, the inner min corresponds to the fact that the oracle may provide a worst-case response to any query; the max corresponds to the fact that the learning algorithm gets to choose the "best" query point a; and the outer min corresponds to the fact that the learner must succeed no matter what subset C' of C the target concept is drawn from. Thus $\hat{\gamma}^C$ is small if there is a large set C' of concepts which are all very similar in that any query eliminates only a few concepts from C'. If this is the case then many membership queries should be required to learn C; formally, we have the following lemma which is a variant of Fact 2 from [12] (the proof is given in Appendix A):

Lemma 4 *Any (classical) exact learning algorithm for C must have sample complexity $\Omega(\frac{1}{\hat{\gamma}^C})$.*

We now develop some tools which will enable us to prove a quantum version of Lemma 4. Let $C' \subseteq C, |C'| \geq 2$ be such that $\gamma^{C'} = \hat{\gamma}^C$ and let $c_1, \ldots, c_{|C'|}$ be a listing of the concepts in C'. Let the *typical concept for* C' be the function $\hat{c} : \{0,1\}^n \to \{0,1\}$ defined as follows: for all $a \in \{0,1\}^n$, $\hat{c}(a)$ is the bit b such that $|C'_{\langle a,b\rangle}| \geq |C'|/2$ (ties are broken arbitrarily; note that a tie occurs only if $\hat{\gamma}^C = 1/2$). The typical concept $\hat{c}$ need not belong to C' or even to C. The *difference matrix* D is the $|C'| \times 2^n$ zero/one matrix where rows are indexed by concepts in C', columns are indexed by strings in $\{0,1\}^n$, and $D_{i,x} = 1$ iff $c_i(x) \neq \hat{c}(x)$. By our choice of C' and the definition of $\hat{\gamma}^C$, each column of D has at most $|C'| \cdot \hat{\gamma}^C$ ones, so the L_1 matrix norm of D is $\|D\|_1 \leq |C'| \cdot \hat{\gamma}^C$.

Our quantum lower bound proof uses ideas which were first introduced by Bennett *et al.* [6]. Let $\mathcal{N}$ be a fixed quantum network architecture and let $U_0, O_1, \ldots, U_{T-1}, O_T, U_T$ be the corresponding sequence of transformations. For $1 \leq t \leq T$ let $|\phi^c_t\rangle$ be the state of the quantum register after the transformations up through U_{t-1} have been performed (we refer to this stage of the computation as time t) if the oracle gate is QMQ_c. As in [6], for $x \in \{0,1\}^n$ let $q_x(|\phi^c_t\rangle)$, the *query magnitude of string x at time t with respect to c,* be the sum of the squared magnitudes in $|\phi^c_t\rangle$ of the basis states which are querying QMQ_c on string x at time t; so if $|\phi^c_t\rangle = \sum_{z \in \{0,1\}^m} \alpha_z |z\rangle$, then

$$q_x(|\phi^c_t\rangle) = \sum_{w \in \{0,1\}^{m-n}} \|\alpha_{xw}\|^2.$$

The quantity $q_x(|\phi^c_t\rangle)$ can be viewed as the amount of amplitude which the network $\mathcal{N}$ invests in the query string x to QMQ_c at time t. Intuitively, the final outcome of $\mathcal{N}$'s computation cannot depend very much on the oracle's responses to queries which have little amplitude invested in them. Bennett *et al.* formalized this intuition in the following theorem ([6], Theorem 3.3):

Theorem 5 *Let $|\phi^c_t\rangle$ be defined as above. Let $F \subseteq \{0,\ldots,T-1\} \times \{0,1\}^n$ be a set of time-string pairs such that $\sum_{(t,x) \in F} q_x(|\phi^c_t\rangle) \leq \frac{\epsilon^2}{T}$. Now suppose the answer to each query instance $(t,x) \in F$ is modified to some arbitrary fixed bit $a_{t,x}$ (these answers need not be consistent*

[2] Note that each O_i only affects the first $n+1$ bits of a basis state. This is without loss of generality since the transformations U_j can "permute bits" of the network.

with any oracle). Let $|\tilde{\phi}_t^c\rangle$ be the state of the quantum register at time t if the oracle responses are modified as stated above. Then $|||\phi_T^c\rangle - |\tilde{\phi}_T^c\rangle| \le \epsilon$.

The following lemma, which is an extension of Corollary 3.4 from [6], shows that no quantum learning algorithm which makes few QMQ queries can effectively distinguish many concepts in C' from the typical concept $\hat{c}$.

Lemma 6 *Fix any quantum network architecture $\mathcal{N}$ which has query complexity T. For all $\epsilon > 0$ there is a set $S \subseteq C'$ of cardinality at most $T^2|C'|\hat{\gamma}^C/\epsilon^2$ such that for all $c \in C' \setminus S$, we have $|||\phi_T^{\hat{c}}\rangle - |\phi_T^c\rangle| \le \epsilon$.*

Proof: Since $|||\phi_t^{\hat{c}}\rangle| = 1$ for all $t = 0, 1, \ldots, T-1$, we have $\sum_{t=0}^{T-1}\sum_{x\in\{0,1\}^n} q_x(|\phi_t^{\hat{c}}\rangle) = T$. Let $q(|\phi_t^{\hat{c}}\rangle) \in \Re^{2^n}$ be the 2^n-dimensional vector which has entries indexed by strings $x \in \{0,1\}^n$ and which has $q_x(|\phi_t^{\hat{c}}\rangle)$ as its x-th entry. Note that the L_1 norm $\|q(|\phi_t^{\hat{c}}\rangle)\|_1$ is 1 for all $t = 0, \ldots, T-1$. For any $c_i \in C'$ let $q_{c_i}(|\phi_t^{\hat{c}}\rangle)$ be defined as $\sum_{x:c_i(x)\neq\hat{c}(x)} q_x(|\phi_t^{\hat{c}}\rangle)$. The quantity $q_{c_i}(|\phi_t^{\hat{c}}\rangle)$ can be viewed as the total query magnitude with respect to $\hat{c}$ at time t of those strings which distinguish c_i from $\hat{c}$. Note that $Dq(|\phi_t^{\hat{c}}\rangle) \in \Re^{|C'|}$ is an $|C'|$-dimensional vector whose i-th element is precisely $\sum_{x:c_i(x)\neq\hat{c}(x)} q_x(|\phi_t^{\hat{c}}\rangle) = q_{c_i}(|\phi_t^{\hat{c}}\rangle)$. Since $\|D\|_1 \le |C'| \cdot \hat{\gamma}^C$ and $\|q(|\phi_t^{\hat{c}}\rangle)\|_1 = 1$, by the basic property of matrix norms we have that $\|Dq(|\phi_t^{\hat{c}}\rangle)\|_1 \le |C'| \cdot \hat{\gamma}^C$, i.e. $\sum_{c_i\in C'} q_{c_i}(|\phi_t^{\hat{c}}\rangle) \le |C'| \cdot \hat{\gamma}^C$. Hence

$$\sum_{t=0}^{T-1}\sum_{c_i\in C'} q_{c_i}(|\phi_t^{\hat{c}}\rangle) \le T|C'| \cdot \hat{\gamma}^C.$$

If we let $S = \{c_i \in C' : \sum_{t=0}^{T-1} q_{c_i}(|\phi_t^{\hat{c}}\rangle) \ge \frac{\epsilon^2}{T}\}$, by Markov's inequality we have $|S| \le T^2|C'|\hat{\gamma}^C/\epsilon^2$. Finally, if $c \notin S$ then $\sum_{t=0}^{T-1} q_c(|\phi_t^{\hat{c}}\rangle) \le \frac{\epsilon^2}{T}$. Theorem 5 then implies that $|||\phi_T^{\hat{c}}\rangle - |\phi_T^c\rangle| \le \epsilon$. ■

Now we can prove our quantum version of Lemma 4.

Theorem 7 *Any quantum exact learning algorithm for C must have sample complexity $\Omega\left(\left(\frac{1}{\hat{\gamma}^C}\right)^{1/2}\right)$.*

Proof: Suppose that $\mathcal{N}$ is a quantum exact learning algorithm for $\mathcal{C}$ which makes at most $T = \frac{1}{64} \cdot \left(\frac{1}{\hat{\gamma}^C}\right)^{1/2}$ quantum membership queries. If we take $\epsilon = \frac{1}{32}$, then Lemma 6 implies that there is a set $S \subset C'$ of cardinality at most $\frac{|C'|}{4}$ such that for all $c \in C' \setminus S$ we have $|||\phi_T^c\rangle - |\phi_T^{\hat{c}}\rangle| \le \frac{1}{32}$. Let c_1, c_2 be any two concepts in $C' \setminus S$. By Fact 3, the probability that $\mathcal{N}$ outputs a circuit equivalent to c_1 can differ by at most $\frac{1}{8}$ if $\mathcal{N}$'s oracle gates are $QMQ_{\hat{c}}$ as opposed to QMQ_{c_1}, and likewise for $QMQ_{\hat{c}}$ versus QMQ_{c_2}. It follows that the probability that $\mathcal{N}$ outputs a circuit equivalent to c_1 can differ by at most $\frac{1}{4}$ if $\mathcal{N}$'s oracle gates are QMQ_{c_1} as opposed to QMQ_{c_2}, but this contradicts the assumption that $\mathcal{N}$ is a quantum exact learning algorithm for C. ■

Known upper bounds on the query complexity of searching a quantum database [9, 23] can easily be used to show that Theorem 7 is tight up to constant factors.

A Lower Bound Based on Concept Class Size. A second reason why a concept class can require many membership queries is its size. Angluin [2] has given the following simple bound, incomparable to the bound of Lemma 4, on the number of classical membership queries required for exact learning (the proof is given in Appendix A):

Lemma 8 *Any classical exact learning algorithm for C must have sample complexity $\Omega(\log|C|)$.*

In this section we prove a variant of this lemma for the quantum model. Our proof uses ideas from [5] so we introduce some of their notation. Let $N = 2^n$. For each concept $c \in C$, let $X^c = (X_0^c, \ldots, X_{N-1}^c) \in \{0,1\}^N$ be a vector which represents c as an N-tuple, i.e. $X_i^c = c(x^i)$ where $x^i \in \{0,1\}^n$ is the binary representation of i. ¿From this perspective we may identify C with a subset of $\{0,1\}^N$, and we may view a QMQ_c gate as a black-box oracle for X^c which maps basis state $|x^i, b, y\rangle$ to $|x^i, b\oplus X_i^c, y\rangle$.

Using ideas from [20, 21], Beals *et al.* have proved the following useful lemma, which relates the query complexity of a quantum network to the degree of a certain polynomial ([5], Lemma 4.2):

Lemma 9 *Let $\mathcal{N}$ be a quantum network that makes T queries to a black-box X, and let $B \subseteq \{0,1\}^m$ be a set of basis states. Then there exists a real-valued multilinear polynomial $P_B(X)$ of degree at most $2T$ which equals the probability that observing the final state of the network with black-box X yields a state from B.*

We use Lemma 9 to prove the following quantum lower bound based on concept class size. (Ronald de Wolf has observed that this lower bound can also be obtained from the results of [19].)

Theorem 10 *Any exact quantum learning algorithm for C must have sample complexity $\Omega\left(\frac{\log|C|}{n}\right)$.*

Proof: Let $\mathcal{N}$ be a quantum network which learns C and has query complexity T. For all $c \in C$ we have the following: if $\mathcal{N}$'s oracle gates are QMQ_c gates, then with probability at least $2/3$ the output of $\mathcal{N}$ is a representation of a Boolean circuit h which computes c. Let $c_1, \ldots, c_{|C|}$ be all of the concepts in C, and let $X^1, \ldots, X^{|C|}$ be the corresponding vectors in $\{0,1\}^N$. For all $i = 1, \ldots, |C|$ let $B_i \subseteq \{0,1\}^m$ be the collection of those basis states which are such that if the final observation performed by $\mathcal{N}$ yields a state from B_i, then the output of $\mathcal{N}$ is a representation of a Boolean circuit which computes c_i. Clearly for $i \neq j$ the sets B_i and B_j are disjoint. By Lemma 9, for each $i = 1, \ldots, |C|$ there is a real-valued multilinear polynomial P_i of degree at most $2T$ such that for all $j = 1, \ldots, |C|$, the value of $P_i(X^j)$ is precisely the probability that the final observation on $\mathcal{N}$ yields a representation of a circuit which computes c_i, provided that the oracle gates are QMQ_{c_j} gates. The polynomials P_i thus have the following properties:

1. $P_i(X^i) \geq 2/3$ for all $i = 1, \ldots, |C|$;
2. For any $j = 1, \ldots, |C|$, we have $\sum_{i \neq j} P_i(X^j) \leq 1/3$ (since the total probability across all possible observations is 1).

Let $N_0 = \sum_{i=0}^{2T} \binom{N}{i}$. For any $X = (X_0, \ldots, X_{N-1}) \in \{0,1\}^N$ let $\tilde{X} \in \{0,1\}^{N_0}$ be the column vector which has a coordinate for each monic multilinear monomial over $X_0, \ldots, X_{N-1}$ of degree at most $2T$. Thus, for example, if $N = 4$ and $2T = 2$ we have $X = (X_0, X_1, X_2, X_3)$ and

$$\begin{aligned}\tilde{X}^t &= (1, X_0, X_1, X_2, X_3, X_0X_1, X_0X_2, \\ &\quad X_0X_3, X_1X_2, X_1X_3, X_2X_3).\end{aligned}$$

If V is a column vector in $\Re^{N_0}$, then $V^t\tilde{X}$ corresponds to the degree-$2T$ polynomial whose coefficients are given by the entries of V. For $i = 1, \ldots, |C|$ let $V_i \in \Re^{N_0}$ be the column vector which corresponds to the coefficients of the polynomial P_i. Let M be the $|C| \times N_0$ matrix whose i-th row is V_i^t; note that multiplication by M defines a linear transformation from $\Re^{N_0}$ to $\Re^{|C|}$. Since $V_i^t\tilde{X}^j$ is precisely $P_i(X^j)$, the product $M\tilde{X}^j$ is a column vector in $\Re^{|C|}$ which has $P_i(X^j)$ as its i-th coordinate.

Now let L be the $|C| \times |C|$ matrix whose j-th column is the vector $M\tilde{X}^j$. A square matrix A is said to be *diagonally dominant* if $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ for all i. Properties (1) and (2) above imply that the transpose of L is diagonally dominant. It is well known that any diagonally dominant matrix must be of full rank (a proof is given in Appendix C). Since L is full rank and each column of L is in the image of M, it follows that the image under M of $\Re^{N_0}$ is all of $\Re^{|C|}$, and hence $N_0 \geq |C|$. Finally, since $N_0 = \sum_{i=0}^{2T} \binom{N}{i} \leq N^{2T}$, we have $T \geq \frac{\log |C|}{2 \log N} = \frac{\log |C|}{2n}$, which proves the theorem. ∎

The lower bound of Theorem 10 is nearly tight as witnessed by the following example: let C be the collection of all 2^n parity functions over $\{0,1\}^n$, so each function in C is defined by a string $a \in \{0,1\}^n$ and $c_a(x) = a \cdot x$. The quantum algorithm which solves the well-known Deutsch-Jozsa problem [17] can be used to exactly identify a and thus learn the target concept with probability 1 from a single query. It follows that the factor of n in the denominator of Theorem 10 cannot be replaced by any function $g(n) = o(n)$.

3.3. Quantum and Classical Exact Learning are Equivalent

We have seen two different reasons why exact learning a concept class can require a large number of classical membership queries: the class may contain many similar concepts (i.e. $\hat{\gamma}^C$ is small), or the class may contain very many concepts (i.e. $\log |C|$ is large). The following lemma, which is a variant of Theorem 3.1 from [24], shows that these are the only reasons why many membership queries may be required (the proof is given in Appendix A).

Lemma 11 *There is an exact learning algorithm for C which has sample complexity $O((\log |C|)/\hat{\gamma}^C)$.*

Combining Theorem 7, Theorem 10 and Lemma 11 we obtain the following relationship between the quantum and classical sample complexity of exact learning:

Theorem 1 *Let $\mathcal{C}$ be any concept class over $\{0,1\}^n$ and let D and Q be such that $\mathcal{C}$ is exact learnable from D classical membership queries or from Q quantum membership queries. Then $D = O(nQ^3)$.*

We note that a QMQ_c oracle can clearly be used to simulate an MQ_c oracle, so $Q \leq D$ as well.

3.4. Discussion

Theorem 1 provides an interesting contrast to several known results for black-box quantum computation. Let F denote the set of all 2^{2^n} functions from $\{0,1\}^n$ to $\{0,1\}$. Beals *et al.* [5] have shown that if $f : F \to \{0,1\}$ is any total function (i.e. $f(c)$ is defined for every possible concept c over $\{0,1\}^n$), then the query complexity of any quantum network which computes f is polynomially related to the number of classical black-box queries required to compute f. Their result is interesting because it is well known [7, 11, 17, 32] that for certain concept classes $C \subset F$ and partial functions $f : C \to \{0,1\}$, the quantum black-box query complexity of f can be exponentially smaller than the classical black-box query complexity.

Our Theorem 1 provides a sort of dual to the results of Beals *et al.*: their bound on query complexity holds only for the fixed concept class F but for any function $f : F \to \{0,1\}$, while our bound holds for any concept class $C \subseteq F$ but only for the fixed problem of exact learning. In general, the problem of computing a function $f : C \to \{0,1\}$ from black-box queries can be viewed as an easier version of the corresponding exact learning problem: instead of having to figure out only one bit of information about the unknown concept c (the value of f), for the learning problem the algorithm must identify c exactly. Theorem 1 shows that for this more demanding problem, unlike the results in [7, 11, 17, 32] there is no way of restricting the concept class C so that learning becomes substantially easier in the quantum setting than in the classical setting.

4. PAC Learning from a Quantum Example Oracle

4.1. The Quantum Example Oracle

Bshouty and Jackson [13] have introduced a natural quantum generalization of the standard PAC-model example oracle. While a standard PAC example oracle $EX(c, \mathcal{D})$ generates each example $\langle x, c(x) \rangle$ with probability $\mathcal{D}(x)$, where $\mathcal{D}$ is a distribution over $\{0,1\}^n$, a *quantum PAC example oracle* $QEX(c, \mathcal{D})$ generates a superposition of all labeled examples, where each labeled example $\langle x, c(x) \rangle$ appears in the superposition with amplitude proportional to the square root of $\mathcal{D}(x)$. More formally, a $QEX(c, \mathcal{D})$ gate maps the initial basis state $|0^n, 0\rangle$ to the state $\sum_{x \in \{0,1\}^n} \sqrt{\mathcal{D}(x)} |x, c(x)\rangle$. (We leave the action of a $QEX(c, \mathcal{D})$ gate undefined on other basis states, and stipulate that any quantum network which includes T $QEX(c, \mathcal{D})$ gates must have all T gates at the "bottom of the circuit," i.e. no gate may occur on any wire between the inputs and any $QEX(c, \mathcal{D})$ gate.) A quantum network with T $QEX(c, \mathcal{D})$ gates is said to be a QEX network with *query complexity* T.

A *quantum PAC learning algorithm for* $\mathcal{C}$ is a family $\{\mathcal{N}_{(n,\epsilon,\delta)} : n \geq 1, 0 < \epsilon, \delta < 1\}$ of QEX networks with the following property: for all $n \geq 1$ and $0 < \epsilon, \delta < 1$, for all $c \in C_n$, for all distributions $\mathcal{D}$ over $\{0,1\}^n$, if the network $\mathcal{N}_{(n,\epsilon,\delta)}$ has all its oracle gates instantiated as $QEX(c, \mathcal{D})$ gates, then with probability at least $1 - \delta$ the network $\mathcal{N}_{(n,\epsilon,\delta)}$ outputs a representation of a circuit h which is an ϵ-approximator to c under $\mathcal{D}$. The *quantum sample complexity* $T(n, \epsilon, \delta)$ of a quantum PAC algorithm is the query complexity of $\mathcal{N}_{(n,\epsilon,\delta)}$.

4.2. Lower Bounds on Classical and Quantum PAC Learning

Throughout this section for ease of notation we omit the subscript n and write C for C_n. We view each concept $c \in C$ as a subset of $\{0,1\}^n$. For $S \subseteq \{0,1\}^n$, we write $\Pi_C(S)$ to denote $\{c \cap S : c \in C\}$, so $|\Pi_C(S)|$ is the number of different "dichotomies" which the concepts in C induce on the points in S. A subset $S \subseteq \{0,1\}^n$ is said to be *shattered* by C if $|\Pi_C(S)| = 2^{|S|}$, i.e. if C induces every possible dichotomy on the points in S. The *Vapnik-Chervonenkis dimension of* C, VC-DIM(C), is the size of the largest subset $S \subseteq \{0,1\}^n$ which is shattered by C.

Well-known results in computational learning theory show that the Vapnik-Chervonenkis dimension of a concept class C characterizes the number of calls to $EX(c, \mathcal{D})$ which are information-theoretically necessary and sufficient to PAC learn C. For the lower bound, the following theorem is a slight simplification of a result due to Blumer *et al.* ([8], Theorem 2.1.ii.b); a proof sketch is given in Appendix A.

Theorem 12 *Let C be any concept class and d = VC-DIM(C). Then any (classical) PAC learning algorithm for $\mathcal{C}$ must have sample complexity $\Omega(d)$.*

We now state a quantum analogue of the classical lower bound given by Theorem 12; the proof uses ideas from error-correcting codes and is given in Appendix B.

Theorem 13 *Let $\mathcal{C}$ be any concept class and d = VC-DIM(C). Then any quantum PAC learning algorithm for $\mathcal{C}$ must have quantum sample complexity $\Omega(\frac{d}{n})$.*

Since the class of parity functions over $\{0,1\}^n$ has VC-dimension n, as in Theorem 10 the n in the denominator of Theorem 13 cannot be replaced by any function $g(n) = o(n)$.

4.3. Quantum and Classical PAC Learning are Equivalent

A well-known theorem due to Blumer *et al.* (Theorem 3.2.1.ii.a of [8]) shows that VC-DIM(C) also upper bounds the number of $EX(c, \mathcal{D})$ calls required for (classical) PAC learning:

Theorem 14 *Let C be any concept class and d = VC-DIM(C). There is a classical PAC learning algorithm for C which has sample complexity $O(\frac{1}{\epsilon} \log \frac{1}{\delta} + \frac{d}{\epsilon} \log \frac{1}{\epsilon})$.*

The proof of Theorem 14 is quite complex so we do not attempt to sketch it. As in Section 3.3, this upper bound along with our lower bound from Theorem 13 together yield:

Theorem 2 *Let $\mathcal{C}$ be any concept class over $\{0,1\}^n$ and let D and Q be such that $\mathcal{C}$ is PAC learnable from D classical examples or from Q quantum examples. Then $D = O(nQ)$.*

We note that a $QEX(c, \mathcal{D})$ oracle can be used to simulate the corresponding $EX(c, \mathcal{D})$ oracle by immediately performing an observation on the QEX gate's outputs[3] (such an observation yields each example $\langle x, c(x) \rangle$ with probability $\mathcal{D}(x)$), and thus $Q \leq D$.

[3] As noted in Section 2.3, intermediate observations during a computation can always be simulated by a single observation at the end of the computation.

5 Quantum versus Classical Efficient Learnability

We have shown that from an information-theoretic perspective, up to polynomial factors quantum learning is no more powerful than classical learning. However, we now observe that the apparant *computational* advantages of the quantum model yield efficient quantum learning algorithms which seem to have no efficient classical counterparts.

A *Blum integer* is an integer $N = pq$ where $p \neq q$ are ℓ-bit primes each congruent to 3 modulo 4. It is widely believed that there is no polynomial-time classical algorithm which can successfully factor a randomly selected Blum integer with nonnegligible success probability.

Kearns and Valiant [26] have constructed a concept class $\mathcal{C}$ whose PAC learnability is closely related to the problem of factoring Blum integers. In their construction each concept $c \in \mathcal{C}$ is uniquely defined by some Blum integer N. Furthermore, c has the property that if $c(x) = 1$ then the prefix of x is the binary representation of N. Kearns and Valiant prove that if there is a polynomial time PAC learning algorithm for $\mathcal{C}$, then there is a polynomial time algorithm which factors Blum integers. Thus, assuming that factoring Blum integers is a computationally hard problem for classical computation, the Kearns-Valiant concept class $\mathcal{C}$ is not efficiently PAC learnable.

On the other hand, in a celebrated result Shor [31] has exhibited a poly(n) size quantum network which can factor any n-bit integer with high success probability. Since each positive example of a concept $c \in \mathcal{C}$ reveals the Blum integer N which defines c, using Shor's algorithm it is easy to obtain an efficient quantum PAC learning algorithm for the Kearns-Valiant concept class. We thus have

Observation 15 *If there is no polynomial-time classical algorithm for factoring Blum integers, then there is a concept class $\mathcal{C}$ which is efficiently quantum PAC learnable but not efficiently classically PAC learnable.*

The hardness results of Kearns and Valiant were later extended by Angluin and Kharitonov [3]. Using a public-key encryption system which is secure against chosen-cyphertext attack (based on the assumption that factoring Blum integers is computationally hard for polynomial-time algorithms), they constructed a concept class $\mathcal{C}$ which cannot be learned by any polynomial-time learning algorithm which makes membership queries. As with the Kearns-Valiant concept class, though, using Shor's quantum factoring algorithm it is possible to construct an efficient quantum exact learning algorithm for this concept class. Thus, for the exact learning model as well, we have:

Observation 16 *If there is no polynomial-time classical algorithm for factoring Blum integers, then there is a concept class $\mathcal{C}$ which is efficiently quantum exact learnable from membership queries but not efficiently classically exact learnable from membership queries.*

Servedio [30] has recently established a stronger separation between the quantum and classical models of exact learning from membership queries than is implied by Observation 16. Using a new construction of pseudorandom functions in conjunction with Simon's quantum oracle algorithm [32], it is shown in [30] that if any one-way function exists then there is a concept class C which is efficiently quantum exact learnable from membership queries but not efficiently classically exact learnable from membership queries.

6 Conclusion and Future Directions

While we have shown that quantum and classical learning are (up to polynomial factors) information-theoretically equivalent, many interesting questions remain about the relationship between *efficient* quantum and classical learnability. It would be interesting to develop efficient quantum learning algorithms for natural concept classes, such as the polynomial-time quantum algorithm of Bshouty and Jackson [13] for learning DNF formulae from uniform quantum examples.

7 Acknowledgements

We thank Ronald de Wolf for helpful comments and observations, and the anonymous referee for helpful suggestions.

References

[1] A. Ambainis. Quantum lower bounds by quantum arguments, *in* "Proc. 32nd ACM Symp. on Theory of Computing," (2000), 636-643. quant-ph/0002066.

[2] D. Angluin. Queries and concept learning, *Machine Learning* **2** (1988), 319-342.

[3] D. Angluin and M. Kharitonov. When won't membership queries help? *J. Comp. Syst. Sci.* **50** (1995), 336-355.

[4] M. Anthony and N. Biggs. *Computational Learning Theory: an Introduction.* Cambridge Univ. Press, 1997.

[5] R. Beals, H. Buhrman, R. Cleve, M. Mosca and R. de Wolf. Quantum lower bounds by polynomials, *in* "Proc. 39th IEEE Symp. on Found. of Comp. Sci.," (1998), 352-361. quant-ph/9802049.

[6] C. Bennett, E. Bernstein, G. Brassard and U. Vazirani. Strengths and weaknesses of quantum computing, *SIAM J. Comput.* **26**(5) (1997), 1510-1523.

[7] E. Bernstein and U. Vazirani. Quantum complexity theory, *SIAM J. Comput.,* **26**(5) (1997), 1411-1473.

[8] A. Blumer, A. Ehrenfeucht, D. Haussler and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis Dimension, *J. ACM* **36**(4) (1989), 929-965.

[9] M. Boyer, G. Brassard, P. Høyer, A. Tapp. Tight bounds on quantum searching, *Forschritte der Physik* **46**(4-5) (1998), 493-505.

[10] G. Brassard, P. Høyer and A. Tapp. Quantum counting, *in* "Proc. 25th ICALP" (1998) 820-831. quant-ph/9805082.

[11] G. Brassard and P. Høyer. An exact quantum polynomial-time algorithm for Simon's problem, *in* "Fifth Israeli Symp. on Theory of Comp. and Systems" (1997), 12-23.

[12] N. Bshouty, R. Cleve, R. Gavaldà, S. Kannan and C. Tamon. Oracles and queries that are sufficient for exact learning, *J. Comput. Syst. Sci.* **52**(3) (1996), 421-433.

[13] N. Bshouty and J. Jackson. Learning DNF over the uniform distribution using a quantum example oracle, *SIAM J. Comput.* **28**(3) (1999), 1136-1153.

[14] H. Buhrman, R. Cleve, R. de Wolf and C. Zalka. Reducing error probability in quantum algorithms, *in* "Proc. 40th IEEE Symp. on Found. of Computer Science," (1999), 358-368. quant-ph/9904019.

[15] H. Buhrman, R. Cleve and A. Wigderson. Quantum vs. classical communication and computation, *in* "Proc. 30th ACM Symp. on Theory of Computing," (1998), 63-68. quant-ph/9802040.

[16] R. Cleve. An introduction to quantum complexity theory, *to appear in* "Collected Papers on Quantum Computation and Quantum Information Theory," ed. by C. Macchiavello, G.M. Palma and A. Zeilinger. quant-ph/9906111.

[17] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation, *Proc. Royal Society of London A,* **439** (1992), 553-558.

[18] R. de Wolf, personal communication, 2000.

[19] E. Farhi, J. Goldstone, S. Gutmann and M. Sipser. How many functions can be distinguished with k quantum queries?, available at `http:/www.arxiv.org/abs/quant-ph/9901012`, 1999.

[20] S. Fenner, L. Fortnow, S. Kurtz and L. Li. An oracle builder's toolkit, *in* "Proc. Eighth Structure in Complexity Theory Conference" (1993), 120-131.

[21] L. Fortnow and J. Rogers. Complexity limitations on quantum computation. *Journal of Comput. and Syst. Sci.* **59**(2) (1999), 240-252.

[22] R. Gavaldà. The complexity of learning with queries, *in* "Proc. Ninth Structure in Complexity Theory Conference" (1994), 324-337.

[23] L. K. Grover. A fast quantum mechanical algorithm for database search, *in* "Proc. 28th Symp. on Theory of Computing" (1996), 212-219.

[24] T. Hegedűs. Generalized teaching dimensions and the query complexity of learning, *in* "Proc. Eighth Conf. on Comp. Learning Theory," (1995), 108-117.

[25] L. Hellerstein, K. Pillaipakkamnatt, V. Raghavan and D. Wilkins. How many queries are needed to learn? *J. ACM* **43**(5) (1996), 840-862.

[26] M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata, *J. ACM* **41**(1) (1994), 67-95.

[27] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory.* MIT Press, 1994.

[28] M. Nielsen and I. Chuang. *Quantum computation and quantum information.* Cambridge University Press, 2000.

[29] J. Ortega. *Matrix Theory: a second course.* Plenum Press, 1987.

[30] R. Servedio. Separating quantum and classical learning, manuscript, 2001.

[31] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.* **26**(5) (1997), 1484-1509.

[32] D. Simon. On the power of quantum computation, *SIAM J. Comput.* **26**(5) (1997), 1474-1483.

[33] L. G. Valiant. A theory of the learnable, *Comm. ACM* **27**(11) (1984), 1134-1142.

[34] J. H. Van Lint. *Introduction to Coding Theory.* Springer-Verlag, 1992.

[35] V.N. Vapnik and A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities, *Theory of Probability and its Applications,* **16**(2) (1971), 264-280.

[36] A.C. Yao. Quantum circuit complexity, *in* "Proc. 34th Symp. on Found. of Comp. Sci." (1993), 352-361.

[37] C. Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A* **60** (1999), 2746–2751.

A Bounds on Classical Sample Complexity

Proof of Lemma 4: Let $C' \subseteq C$, $|C'| \geq 2$ be such that $\gamma^{C'} = \hat{\gamma}^C$. Consider the following adversarial strategy for answering queries: given the query string a, answer the bit b which maximizes $\gamma^{C'}_{\langle a,b\rangle}$. This strategy ensures that each response eliminates at most a $\gamma^{C'}_a \leq \gamma^{C'} = \hat{\gamma}^C$ fraction of the concepts in C'. After $\frac{1}{2\hat{\gamma}^C} - 1$ membership queries, fewer than half of the concepts in C' have been eliminated, so at least two concepts have not yet been eliminated. Consequently, it is impossible for A to output a hypothesis which is equivalent to the correct concept with probability greater than $1/2$. (Lemma 4) ■

Proof of Lemma 8: Consider the following adversarial strategy for answering queries: if $C' \subseteq C$ is the set of concepts which have not yet been eliminated by previous responses to queries, then given the query string a, answer the bit b such that $\gamma^{C'}_{\langle a,b\rangle} \geq \frac{1}{2}$. Under this strategy, after $\log |C| - 1$ membership queries at least two possible target concepts will remain. (Lemma 8) ■

Proof of Lemma 11: Consider the following learning algorithm A: at each stage in its execution, if C' is the set of concepts in C which have not yet been eliminated by previous responses to queries, algorithm A's next query string is the string $a \in \{0,1\}^n$ which maximizes $\gamma^{C'}_a$. By following this strategy, each query response received from the oracle must eliminates at least a $\gamma^{C'}$ fraction of the set C', so with each query the size of the set of possible target concepts is multiplied by a factor which is at most $1 - \gamma^{C'} \leq 1 - \hat{\gamma}^C$. Consequently, after $O((\log |C|)/\hat{\gamma}^C)$ queries, only a single concept will not have been eliminated; this concept must be the target concept, so A can output a hypothesis h which is equivalent to c. (Lemma 11) ■

Proof Sketch for Theorem 12: The idea behind Theorem 12 is to consider the distribution $\mathcal{D}$ which is uniform over some shattered set S of size d and assigns zero weight to points outside of S. Any learning algorithm which makes only $d/2$ calls to $EX(c, \mathcal{D})$ will have no information about the value of c on at least $d/2$ points in S; moreover, since the set S is shattered by C, any labeling is possible for these unseen points. Since the error of any hypothesis h under $\mathcal{D}$ is the fraction of points in S where h and the target concept disagree, a simple analysis shows that no learning algorithm which perform only $d/2$ calls to $EX(c, \mathcal{D})$ can have high probability (e.g. $1 - \delta = 2/3$) of generating a low-error hypothesis (e.g. $\epsilon = 1/10$). (Theorem 12) ■

B Proof of Theorem 13

Let $S = \{x^1, \ldots, x^d\}$ be a set which is shattered by C and let $\mathcal{D}$ be the distribution which is uniform on S and assigns zero weight to points outside S. If $h : \{0,1\}^n \rightarrow \{0,1\}$ is a Boolean function on $\{0,1\}^n$, we say that the *relative distance of h and c on S* is the fraction of points in S on which h and c disagree. We will prove the following result which is stronger than Theorem 13: Let $\mathcal{N}$ be a quantum network with QMQ gates such that for all $c \in C$, if $\mathcal{N}$'s oracle gates are QMQ_c gates, then with probability at least $2/3$ the output of $\mathcal{N}$ is a hypothesis h such that the relative distance of h and c on S is at most $1/10$. We will show that such a network $\mathcal{N}$ must have query complexity at least $\frac{d}{12n}$. Since any QEX network with query complexity T can be simulated by a QMQ network with query complexity T, taking $\epsilon = 1/10$ and $\delta = 1/3$ will prove Theorem 13.

The argument is a modification of the proof of Theorem 10 using ideas from error correcting codes. Let $\mathcal{N}$ be a quantum network with query complexity T which satisfies the following condition: for all $c \in C$, if $\mathcal{N}$'s oracle gates are QMQ_c gates, then with probability at least $2/3$ the output of $\mathcal{N}$ is a representation of a Boolean circuit h such that the relative distance of h and c on S is at most $1/10$. By the well-known Gilbert-Varshamov bound from coding theory (see, e.g., Theorem 5.1.7 of [34]), there exists a set $s^1, \ldots, s^A$ of d-bit strings such that for all $i \neq j$ the strings s^i and s^j differ in at least $d/4$ bit positions, where

$$A \geq \frac{2^d}{\sum_{i=0}^{d/4-1} \binom{d}{i}} \geq \frac{2^d}{\sum_{i=0}^{d/4} \binom{d}{i}} \geq 2^{d(1-H(1/4))} > 2^{d/6}.$$

(Here $H(p) = -p \log p - (1-p) \log(1-p)$ is the binary entropy function.) For each $i = 1, \ldots, A$ let $c_i \in C$ be a concept such that the d-bit string $c_i(x^1) \cdots c_i(x^d)$ is s^i (such a concept c_i must exist since the set S is shattered by C).

For $i = 1, \ldots, A$ let $B_i \subseteq \{0,1\}^m$ be the collection of those basis states which are such that if the final observation performed by $\mathcal{N}$ yields a state from B_i, then the output of $\mathcal{N}$ is a hypothesis h such that h and c_i have relative distance at most $1/10$ on S. Since each pair of concepts c_i, c_j has relative distance at least $1/4$ on S, the sets B_i and B_j are disjoint for all $i \neq j$.

As in Section 3.2 let $N = 2^n$ and let $X^j = (X^j_0, \ldots, X^j_{N-1}) \in \{0,1\}^n$ where X^j is the N-tuple representation of the concept c_j. By Lemma 9, for each $i = 1, \ldots, A$ there is a real-valued multilinear polynomial P_i of degree at most $2T$ such that for all $j = 1, \ldots, A$, the value of $P_i(X^j)$ is precisely the probability that the final observation on $\mathcal{N}$ yields a state from B_i provided that the oracle gates are QMQ_{c_j} gates. Since, by assumption, if c_i is the target concept then with probability at least $2/3$ $\mathcal{N}$ generates a hypothesis which has relative distance at most $1/10$

from c_i on S, the polynomials P_i have the following properties:

1. $P_i(X^i) \geq 2/3$ for all $i = 1, \ldots, A$;
2. For any $j = 1, \ldots, A$ we have that $\sum_{i \neq j} P_i(X^j) \leq 1/3$ (since the B_i's are disjoint and the total probability across all observations is 1).

Let N_0 and $\tilde{X}$ be defined as in the proof of Theorem 10. For $i = 1, \ldots, A$ let $V_i \in \Re^{N_0}$ be the column vector which corresponds to the coefficients of the polynomial P_i, so $V_i^t \tilde{X} = P_i(X)$. Let M be the $A \times N_0$ matrix whose i-th row is the vector V_i^t, so multiplication by M is a linear transformation from $\Re^{N_0}$ to $\Re^A$. The product $M\tilde{X}^j$ is a column vector in $\Re^A$ which has $P_i(X)$ as its i-th coordinate.

Now let L be the $A \times A$ matrix whose j-th column is the vector $M\tilde{X}^j$. As in Theorem 10 we have that the transpose of L is diagonally dominant, so L is of full rank and hence $N_0 \geq A$. Since $A \geq 2^{d/6}$ we thus have that $T \geq \frac{d/6}{2\log_2 N} = \frac{d}{12n}$, and the theorem is proved. (Theorem 13) ■

C A diagonally dominant matrix has full rank

This fact follows from the following theorem (see, e.g., Theorem 6.1.17 of [29]).

Theorem 17 (Gershgorin's Circle Theorem) *Let A be a real or complex-valued $n \times n$ matrix. Let S_i be the disk in the complex plane whose center is a_{ii} and whose radius is $r_i = \sum_{j \neq i} |a_{ij}|$. Then every eigenvalue of A lies in the union of the disks $S_1, \ldots, S_n$.*

The proof is well known: if λ is an eigenvalue of A which has corresponding eigenvector $x = (x_1, \ldots, x_n)$, then since $Ax = \lambda x$ we have

$$(\lambda - a_{ii})x_i = \sum_{j \neq i} a_{ij} x_j \ \text{ for } i = 1, \ldots, n.$$

Without loss of generality we may assume that $||x||_\infty = 1$, so $|x_k| = 1$ for some k and $|x_j| \leq 1$ for $j \neq k$. Thus

$$|\lambda - a_{kk}| = |(\lambda - a_{kk})x_k| \leq \sum_{j \neq k} |a_{kj}||x_j| \leq \sum_{j \neq k} |a_{kj}|$$

and hence λ is in the disk S_k.

For a diagonally dominant matrix the radius r_i of each disk S_i is less than its distance from the origin, which is $|a_{ii}|$. Hence 0 cannot be an eigenvalue of a diagonally dominant matrix, so the matrix must have full rank.