

How Many Query Superpositions Are Needed to Learn?

Jorge Castro

Software Department. Universitat Politècnica de Catalunya. Campus Nord, 08034
Barcelona, Spain. castro@lsi.upc.edu

Abstract. This paper introduces a framework for quantum exact learning via queries, the so-called quantum protocol. It is shown that usual protocols in the classical learning setting have quantum counterparts. A combinatorial notion, the general halving dimension, is also introduced. Given a quantum protocol and a target concept class, the general halving dimension provides lower and upper bounds on the number of queries that a quantum algorithm needs to learn. For usual protocols, the lower bound is also valid even if only involution oracle teachers are considered. Under some protocols, the quantum upper bound improves the classical one. The general halving dimension also approximates the query complexity of ordinary randomized learners. From these bounds we conclude that quantum devices can allow moderate improvements on the query complexity. However, any quantum polynomially query learnable concept class must be also polynomially learnable in the classical setting.

1 Introduction

A central topic in quantum computation concerns the query complexity of oracle machines. Often it is assumed that a quantum device can get partial information about an unknown function making some type of oracle calls. The broad goal is to take advantage of quantum mechanic effects in order to improve the number of queries (or oracle calls) that an ordinary algorithm needs to find out some characteristic of the hidden function. In some cases it has been proved that exponentially fewer black-box oracle calls (also called membership queries) are required in the quantum model, see for instance [13, 18]. On the other hand, there are tasks that do not accept huge improvements on the query complexity. For example, it is known that the quadratic speedup of Grover's quantum algorithm for database search is optimal [14]. Furthermore, quite general lower bounds on the number of oracle interactions have been also obtained [1, 7, 9].

Quantum concept learning can be seen as a special case of this type of research where the goal of the algorithm is to figure out which the hidden function is. Here several results are known. Bshouty and Jackson [12] define a quantum version of the PAC model and provide a quantum learning algorithm for DNF that does not require memberships, a type of queries used by its classical counterpart. Servedio and Gortler [17] show lower bounds on the number of oracle calls required to learn on the quantum PAC setting and on the more

demanding scenario of exact learning from membership queries. For both specific learning settings they conclude that dramatic improvements on the number of oracle interactions are not possible. Ambainis et al. [2] and Atici and Servedio [4] give non-trivial upper bounds for quantum exact learning from membership queries. Finally, Hunziker et al. [16] show a general technique for quantum learning from memberships and restricted equivalences that is shown to need, in a couple of specific cases, less number of queries than is possible classically.

This paper has two goals. The first one is to introduce a general framework for quantum exact learning via queries which sets when a class of queries can be considered to define a learning game played by quantum devices. We note that, as far as we know, the only queries that have been used in the literature have been memberships [2, 4, 16, 17] and restricted equivalences [16]. This contrasts with the classical setting where a rich variety of queries have been considered, see for instance Angluin [3]. The second goal is to study the number of queries (or query complexity) required by exact learners. Our aim is to obtain lower and upper bounds on the query complexity that are valid under any choice of queries defining the learning game.

According to the first goal, we introduce in Sect. 3 the quantum protocol concept, a notion that allows us to define a learning game played by quantum machines where popular queries from the classical setting, as memberships, equivalences, subsets and others defined in [3] have natural quantum counterparts. Specific quantum protocols for these queries are presented. Learning games defined by quantum protocols for memberships and memberships and restricted equivalences agree with learning settings present in the literature [2, 4, 16, 17].

With respect to the second goal, we define in Sect. 4 a combinatorial function, the general halving dimension, GHdim , having some nice features. In the quantum learning scenario, we show a lower bound for the query complexity in terms of GHdim that is valid for any quantum protocol and for any target concept class (Theorem 9). We also show a generic quantum algorithm that achieves learning on many quantum protocols and provides an upper bound for the query complexity in terms of GHdim (Theorem 14). These lower and upper bounds extend the previous ones in [4, 17] for the specific protocol of membership queries. In the classical learning model, we prove that GHdim approximates the query complexity of randomized learners (Theorems 11 and 15). This characterization extends the previous ones provided by Simon [19] for the specific ordinary protocols of membership and membership and equivalence queries.

From previous results we state in Sect. 5 the following conclusion. Given an arbitrary set of queries, quantum learners can allow some gain on the number of queries needed to learn but huge improvements are not possible. Specifically, we show that any quantum polynomially query learnable concept class must be also polynomially learnable in the ordinary setting (Theorem 16). This fact was only known for membership queries [17].

2 Preliminaries

2.1 Basic Definitions

Given a complex number α , we denote by α^* its complex conjugate and by $|\alpha|$ its module. For complex vectors v and w , the l_2 -norm (Euclidean norm) of v is expressed by $\|v\|$, the l_1 -norm by $\|v\|_1$ and the inner product of v and w by $\langle v|w\rangle$. Note that $\|v\| = \langle v|v\rangle^{1/2}$. Abusing notation, we also denote the cardinality of a set A by $|A|$. For $b, d \in \{0, 1\}$ we write $b \oplus d$ to denote $b + d \pmod{2}$. For n -bit strings $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ we write $x \oplus y$ to denote $(x_1 \oplus y_1, \dots, x_n \oplus y_n)$. The set of all Boolean functions on $\{0, 1\}^n$ is denoted by B_n . A *concept* f is a function of B_n . Equivalently, a concept f can be viewed as the subset $\{x \in \{0, 1\}^n \mid f(x) = 1\}$. A *concept class* C is a subset of B_n .

2.2 Classical Exact Learning

In query learning two players, the *learner* and the *teacher*, play a game. The learner is a (classical) randomized algorithm and the teacher is an oracle function. Some concept class C (the *target* concept class) is known to both players and the teacher chooses a concept in C (the target concept) that is unknown to the learner. The goal of the learner is to find out what concept is, asking the teacher some type of queries.

A query is a question that the learner poses to the teacher. The most popular in the literature are *membership queries* and *equivalence queries*. Other type of queries, as subsets, supersets and restricted equivalences, have been defined, see [3]. In general, the setting of the learning game is complete when the learning *protocol* is defined. The protocol is the agreement about which the admissible queries are and, for each target concept, which the possible answers for such queries are. Answers provide a property of the target. A teacher is valid for the target concept f and the protocol P if it replies to each query q choosing one of the admissible answers in P for q and f .

A concept class C is learnable with k queries under protocol P if there is a randomized learning algorithm L such that for any $f \in C$ and for any valid teacher T that answers with respect to f using P , with probability at least $2/3$ L outputs a circuit h such that $h(x) = f(x)$ for all $x \in \{0, 1\}^n$ after at most k interactions with T . For a class $C \subseteq B_n$ and a protocol P , the *query complexity*, is the smallest k such that C is learnable with k queries under P .

2.3 Quantum Computation

Detailed descriptions of quantum Turing machines and quantum oracle computations are provided in [9, 10]. In spite of assuming the reader is familiar with basic aspects of quantum computers, we provide below a short summary of essential elements.

To each quantum Turing machine M corresponds an inner-product vector space S . The vectors of S are *superpositions* (i.e. finite complex linear combinations) of configurations of M . The complex coefficients defining a vector of S are

called *amplitudes*. The inner-product is defined by given an orthonormal basis for S , the vectors of this basis are the configurations of M . The time evolution operator of a quantum Turing machine M is determined by an unitary matrix U_M , which defines a linear operator on S that conserves the distance.

At step j of the computation of M , the time evolution operator U_M is applied to a superposition of configurations (a vector $|v_j\rangle$ of S). The initial superposition $|v_0\rangle$ is the linear combination of configurations having all amplitude value 0 except the only one corresponding to the initial configuration of the machine that has value 1.

A quantum Turing machine M finishes at step t if the corresponding superposition $|v_t\rangle$ only has nonzero amplitudes on final configurations (those whose state is a final one) and previous superpositions $|v_j\rangle$ where $j < t$ give amplitude zero to each final configuration. Let us assume that M finishes at step t and that $|v_t\rangle = \sum_x \alpha_x |x\rangle$ is the corresponding superposition. Now the machine M chooses to be in a single configuration rather than in a superposition of configurations making an *observation* (or *measurement*). The superposition is then changed so that a single configuration has amplitude 1 and all others are 0. Formally, the observation operation provides configuration $|x\rangle$ with probability $|\alpha_x|^2$. Note that $\sum_x |\alpha_x|^2 = 1$ because $|v_t\rangle$ has norm 1 (it is obtained by applying an unitary operator to an initial superposition $|v_0\rangle$ that has norm 1).

Oracle Quantum Turing Machine We follow definitions in [9]. An oracle quantum Turing machine has a special query tape (that has to accomplish some rules of behaviour, see [9]) and two distinguished internal states: a pre-query state p_1 and a post-query state p_2 . A query is executed whenever the machine enters the pre-query state. In this case, it applies a fixed unitary operator U to the current contents $|q\rangle$ of the query tape, replacing it by $U|q\rangle$. In order to ensure that a single machine cycle ought not to make infinite changes in the tape, we require that $U|q\rangle$ have amplitude zero on all but finitely many basis vectors. The use of this kind of unitary oracles still provide unitary time evolution for, in other aspects, well-defined quantum Turing machines. Another natural restriction one may wish to impose upon U is that it be an involution, $U^2 = I$, so that the effect of an oracle call can be undone by a further call on the same oracle. This may be crucial to allow proper interference to take place.

3 Quantum Exact Learning

The learning game is similar to the classical one but now the learner is a quantum algorithm and the teacher is a quantum oracle function. The game is completely defined when the learning protocol is provided.

3.1 Quantum protocols

We show here how to adapt the learning protocol notion [5, 6] to the quantum setting. A protocol P specifies which the admissible queries are and, for each

query, which the valid answers are. Queries belong to a finite set Q , answers are from a finite set A and P is a subset of $Q \times A$. To each tuple (q, a) of P corresponds a subset of B_n so-called *consistent set* and denoted by σ_q^a . Functions in σ_q^a are said to be consistent with tuple (q, a) . In the learning game defined by protocol P , answer a to query q provides the information that the target function belongs to σ_q^a . We also denote by Σ_q the set of consistent sets defined by the valid answers to query q , so $\Sigma_q = \{\sigma_q^a \mid a \text{ is an answer for } q \text{ in } P\}$.

Discussion above encompasses any type of protocol, classical or quantum. A distinguishing feature of quantum protocols is that different queries can provide the same information. This is an useless characteristic in the classical scenario, but it makes possible to define teachers that as quantum oracles, in addition to be unitary operators are also involutions, a property that one may wish to impose to a quantum oracle to allow proper interference to take place, as we have noted in Section 2.3. Queries showing the same information are said to be *equivalent*. Formally, given a protocol $P \subseteq Q \times A$, queries q_i and q_j are equivalent if their respective sets of consistent function sets defined by their (respective) valid answers coincide, in short $\Sigma_{q_i} = \Sigma_{q_j}$. The equivalence class of query q is denoted by $[q]$ and the set of equivalence classes by $[Q]$.

Definition 1. *A subset P of $Q \times A$ defines a quantum protocol iff P satisfies the following requirements,*

1. *Completeness: Given a query q of Q and a function f in B_n there exists an answer a such that (q, a) is a tuple of P and function f is consistent with (q, a) (in short, $f \in \sigma_q^a$).*
2. *If q_i and q_j are non-equivalent queries then they do not share any valid answer.*
3. *If a is a valid answer for two different queries q_i and q_j then the consistent sets of (q_i, a) and (q_j, a) , respectively $\sigma_{q_i}^a$ and $\sigma_{q_j}^a$, are different.*

The completeness requirement is the only one necessary in order to define a classical protocol. Its justification can be found in [5, 6]. On the other hand, last two requirements in Definition 1 are specific for the quantum setting and they impose some compatible behaviour of P with respect to the equivalence relation it defines on Q . Both are considered by technical convenience (see Lemmas 3 and 4 below).

As first example we consider the protocol consisting of quantum membership queries (or quantum black-box oracle calls). A quantum black-box oracle for function f in B_n transforms $(x, b) \in \{0, 1\}^n \times \{0, 1\}$ to $(x, b \oplus f(x))$. Thus, in the corresponding protocol the set of queries and the set of answers are both $\{0, 1\}^n \times \{0, 1\}$. Valid answers to query (x, b) are $(x, 0)$ and $(x, 1)$. So, tuples of the protocol are $((x, b), (x, b'))$ for all x in $\{0, 1\}^n$ and for all b and b' in $\{0, 1\}$. The consistent set of answer (x, b') to query (x, b) is the set of functions that evaluate to $b' \oplus b$ on x . Queries (x, b) and (y, d) are equivalent whenever $x = y$. Note that the quantum protocol requirements are trivially satisfied.

A quantum version of the classical equivalence query protocol can be defined as follows. Given a hypothesis class H , where H is a subset of B_n , queries and

answers are tuples (h, x, b) belonging to $H \times \{0, 1\}^n \times \{0, 1\}$. Valid answers to query (h, x, b) are $(h, x \oplus y, b)$ for any $y \in \{0, 1\}^n$ and $(h, x, 1 \oplus b)$. The consistent set corresponding to answer $(h, x \oplus y, b)$ are those Boolean functions f such that $f(y) \neq h(y)$. The consistent set of answer $(h, x, 1 \oplus b)$ has only a single element, the function h . Note that queries (h, x, b) and (g, z, d) are equivalent whenever $h = g$. It is straightforward to see that this defines a quantum protocol. Quantum protocols for subsets, restricted equivalences, memberships and equivalences, and other type of popular queries can be defined in a similar way.

3.2 Quantum Teachers

Let $P \subseteq Q \times A$ be a quantum protocol. We associate to the set of queries Q a Hilbert space S_Q defined as follows. Vectors of S_Q are superpositions of query vectors $|q\rangle$ where q is a query of Q . The inner product of S_Q is the one defined by considering the set of query vectors $\{|q\rangle \mid q \in Q\}$ as an orthonormal basis. In a similar way, we also define a Hilbert space S_A corresponding to the set of answers A .

Let f be a Boolean function. A *quantum teacher* for f under protocol P is an unitary operator T transforming each basis query vector $|q\rangle$ to a superposition in S_A of valid answers according to P that are consistent with f . Quantum teacher T for f is said to be a *permutation teacher* whenever it transforms each basis query $|q\rangle$ to a consistent basis answer $|a\rangle$. When $S_Q = S_A$ and the quantum teacher operator T holds that $T^2 = I$, we say that T is an *involution teacher*. Involution teachers shall correspond with involution oracle gates.

We highlight that classical deterministic teachers for memberships, equivalences, subsets and other popular queries trivially define corresponding permutation teachers in the quantum setting. Note that they are also involution teachers.

3.3 Query Complexity

A superposition $|\phi\rangle$ of an oracle quantum machine is said to be a *query superposition* if there is a configuration with nonzero amplitude in $|\phi\rangle$ whose state is the pre-query one. Let P be a quantum protocol. A concept class $C \subseteq B_n$ is learnable under protocol P with m query superpositions if there exists an oracle quantum Turing machine L –so-called learner– such that for any target function f in C and for any quantum teacher T for f under P :

1. L^T gets a final superposition and with probability at least $2/3$, outputs a circuit for f .
2. The computation of L^T yields at most m query superpositions.

For target class C and quantum protocol P we define the *quantum query complexity*, $QC(C, P)$, as the smallest m such that C is learnable with m query superpositions under P . We note that this query complexity notion is consistent with the definition given in Beals et al. [7] (see also Servedio et al. [17]) for quantum networks.

3.4 Answering Schemes

Let $P \subseteq Q \times A$ be a quantum protocol.

Definition 2. A subset \mathcal{T} of P is said to be an answering scheme if:

1. For any query $q \in Q$ there is exactly one answer a such that (q, a) belongs to \mathcal{T} .
2. If (q_i, a_i) and (q_j, a_j) are tuples of \mathcal{T} and q_i and q_j are equivalent queries then (q_i, a_i) and (q_j, a_j) define the same consistent set of Boolean functions.

The following lemma is an immediate consequence of the quantum protocol and the answering scheme definitions.

Lemma 3. Answers of an answering scheme are all different.

Thus, observe that answering schemes extend naturally to unitary transformations from S_Q to S_A (see Sect. 3.2 above) and they can be considered as quantum oracle functions. However, for an answering scheme \mathcal{T} it is possible that there is no function in B_n consistent with all tuples in \mathcal{T} . This contrasts with the quantum teacher notion introduced above where there is always a consistent Boolean function with all teacher answers. As we will see later, answering schemes have an adversary role in our arguments in Section 4.2.

Let L be a quantum learner under protocol P and let \mathcal{T} be an answering scheme of P . We consider the computation of L when oracle calls are solved according to \mathcal{T} and we denote by $L^{\mathcal{T}}$ the resulting quantum oracle machine. Let $|\phi\rangle$ be a valid superposition of $L^{\mathcal{T}}$. We define the *query magnitude* of q in $|\phi\rangle$, denoted by $w_q(|\phi\rangle)$, as the weight of query q in superposition $|\phi\rangle$; formally, $w_q(|\phi\rangle) = \sum_c |\alpha_c|^2$ where the sum extends over configurations c querying q and α_c denotes the amplitude of c in $|\phi\rangle$. We naturally extend the query magnitude concept to query classes: $w_{[q]}(|\phi\rangle)$ is the sum of query magnitudes $w_{q'}(|\phi\rangle)$ where q' is any query equivalent to q .

For the specific case of membership queries Bennet *et al.* (Theorem 3.3 in [9]) showed that the final outcome of L 's computations cannot depend very much on the oracle's answers to queries of little magnitude. We extend this result to any quantum protocol in Theorem 5 below. We provide some proof details for two reasons. First, we think that it is a non-trivial extension of the original theorem statement. Second, as we point out later, there is an incorrect assertion in the proof shown in [9]. In the rest of this section, we assume an arbitrary underlying quantum protocol is given.

Lemma 4. Let $|\phi\rangle$ be a valid superposition of $L^{\mathcal{T}}$. Let $G \subseteq [Q]$ be a set of query classes and let \mathcal{T} be any answering scheme that agrees with \mathcal{T} on any query q such that $[q] \notin G$. Let U and \tilde{U} be, respectively, the unitary time operators of $L^{\mathcal{T}}$ and $L^{\tilde{\mathcal{T}}}$. Then, $\|U|\phi\rangle - \tilde{U}|\phi\rangle\|^2 \leq 4 \sum_{[q] \in G} w_{[q]}(|\phi\rangle)$.

Proof. Let $|E\rangle = U|\phi\rangle - \tilde{U}|\phi\rangle$ be the error vector. Assume that $|\phi\rangle = \sum_{c \in I^G} \alpha_c c + |\varphi\rangle$ where I^G is the set of configurations querying some query

equivalent to those defined by G and $|\varphi\rangle$ is a superposition of configurations with no query in G . Then,

$$\begin{aligned} \|E\|^2 &= \sum_{c,d \in I^G} \alpha_c \alpha_d^* \langle Uc|Ud\rangle + \sum_{c,d \in I^G} \alpha_c \alpha_d^* \langle \tilde{U}c|\tilde{U}d\rangle \\ &\quad - \sum_{c,d \in I^G} \alpha_c \alpha_d^* \langle Uc|\tilde{U}d\rangle - \sum_{c,d \in I^G} \alpha_c \alpha_d^* \langle \tilde{U}c|Ud\rangle. \end{aligned}$$

In this expression, by orthogonality the first two summands are both equal to $\sum_{[q] \in G} w_{[q]}(|\phi\rangle)$. For the last two summands observe that all scalar products are zero except for those configurations c and d such that $Uc = \tilde{U}d$. Given a configuration c_0 there is at most one d_0 where this equality happens because the answers of an answering scheme are all different, see Lemma 3. Thus, denoting by J the set of configuration pairs (c_0, d_0) such that $c_0, d_0 \in I^G$ and $Uc_0 = \tilde{U}d_0$, it holds that

$$\begin{aligned} \left| \sum_{c,d \in I^G} \alpha_c \alpha_d^* \langle Uc|\tilde{U}d\rangle + \sum_{c,d \in I^G} \alpha_c \alpha_d^* \langle \tilde{U}c|Ud\rangle \right| &= \left| \sum_{(c_0, d_0) \in J} \alpha_{c_0} \alpha_{d_0}^* + \sum_{(c_0, d_0) \in J} \alpha_{d_0} \alpha_{c_0}^* \right| = \\ \left| \sum_{(c_0, d_0) \in J} 2\text{Re}(\alpha_{c_0} \alpha_{d_0}^*) \right| &\leq \sum_{(c_0, d_0) \in J} 2|\alpha_{c_0}| |\alpha_{d_0}^*| \leq \sum_{(c_0, d_0) \in J} |\alpha_{c_0}|^2 + |\alpha_{d_0}|^2 \leq 2 \sum_{[q] \in G} w_{[q]}(|\phi\rangle). \end{aligned}$$

Therefore, $\|E\|^2 \leq 4 \sum_{[q] \in G} w_{[q]}(|\phi\rangle)$. \square

We note that the proof of Theorem 3.3 in [9] states (see first line in the last paragraph of the proof) that $\|E\|^2 = 2 \sum_{[q] \in G} w_{[q]}(|\phi\rangle)$, that is a better characterization than the inequality given by Lemma 4. However, a counterexample for this equality can be provided under the membership query protocol (which is the protocol considered in [9]). Interested readers can download such counterexample at <http://www.lsi.upc.edu/~castro/counter.pdf>.

Theorem 5. *Let $|\phi_i\rangle$ be the superposition of $L^{\mathcal{T}}$ at time i . Let $\epsilon > 0$. Let $F \subseteq \{0, \dots, t-1\} \times [Q]$ be a set of time-query class pairs such that $\sum_{(i, [q]) \in F} w_{[q]}(|\phi_i\rangle) \leq \frac{\epsilon^2}{4t}$. For each i , let $\tilde{\mathcal{T}}_i$ be any answering scheme that agrees with \mathcal{T} on any query q such that $(i, [q]) \notin F$. Let $|\tilde{\phi}_t\rangle$ be the time t superposition that L will get if the answer to each query instance $(i, [q]) \in F$ is modified according to $\tilde{\mathcal{T}}_i$. Then, $\| |\phi_t\rangle - |\tilde{\phi}_t\rangle \| < \epsilon$.*

4 The Query Complexity of Exact Learners

4.1 The General Halving Dimension

Let $C \subseteq B_n$ be a concept class and let P be a protocol. We associate the parameter $\text{ghdim}(V, P)$ to each subset V of C with $|V| > 1$. This parameter is

the smallest non-negative integer d satisfying the following predicate: for any answering scheme \mathcal{T} from P there exists a subset $S \subseteq \mathcal{T}$ of cardinality d such that at most half of the functions in V are consistent with all tuples in S . When there is no integer d satisfying the predicate, $\text{ghdim}(V, P)$ is defined to be ∞ .

Definition 6. *The general halving dimension of C under P , $\text{GHdim}(C, P)$, is the maximum of parameters $\text{ghdim}(V, P)$. Thus,*

$$\text{GHdim}(C, P) = \max\{\text{ghdim}(V, P) \mid V \subseteq C \wedge |V| > 1\}.$$

The general halving dimension has two ancestors. One is the general dimension concept — which is in turn an extension of the certificate size notion introduced by Hellerstein *et al.* [15]— that is shown to be a nice characterization of the query complexity of deterministic learners in the ordinary learning scenario (see [5]). The other one is the halving complexity notion defined by Simon [19], that approximates the query complexity of randomized learners in the classical setting. We prove below several bounds of the query complexity in terms of the general halving dimension as much for quantum protocols as for classical ones.

4.2 A Lower Bound for the Quantum Query Complexity

Lemma 7. *Let us assume $\text{GHdim}(C, P) > l \geq 1$. There exists a set of concepts $V \subseteq C$ with $|V| > 1$ and an answering scheme \mathcal{T} such that for any tuple $(q, a) \in \mathcal{T}$ less than $\frac{|V|}{l}$ concepts from V are not consistent with (q, a) .*

Proof. For the sake of contradiction suppose that for each subset V of C with $|V| > 1$ and for any answering scheme \mathcal{T} there exists a tuple $(q, a) \in \mathcal{T}$ such that at least $\frac{|V|}{l}$ concepts from V are not consistent with (q, a) . Fix $V = V_0$ and let \mathcal{T} be an answering scheme. Thus, it corresponds to V_0 a tuple $(q_0, a_0) \in \mathcal{T}$ such that at least $\frac{|V_0|}{l}$ concepts from V_0 are not consistent with (q_0, a_0) . Let V_1 be the subset of V_0 consistent with (q_0, a_0) . By assumption, $|V_1| \leq |V_0|(1 - 1/l)$. We repeat this process with V_1 instead of V_0 and so on and so forth. After l iterations we get a subset V_l of V with $|V_l| \leq |V|/2$. This implies that $\text{ghdim}(V, P) \leq l$. \square

Let l be such that $1 \leq l < \text{GHdim}(C, P)$ and let V and \mathcal{T} be respectively the subset of C and the answering scheme promised by Lemma 7. Inspired by Servedio *et al.* [17], we define the *difference matrix* M as the $|V| \times |Q|$ zero/one matrix where rows are indexed by concepts in V , columns are indexed by queries in Q , and $M_{f,q} = 1$ iff the Boolean function f is not consistent with the answer a of q in \mathcal{T} . By our choice of V and \mathcal{T} , each column of M has less than $\frac{|V|}{l}$ ones. Thus, the l_1 matrix norm of M is $\|M\|_1 < \frac{|V|}{l}$. The following lemma, which is a technical generalization of Lemma 6 from [17], shows that no quantum learning algorithm L with small query complexity can effectively distinguish many concepts in V .

Lemma 8. *Let L be a quantum learner with query complexity m . Let $\epsilon > 0$. There are a set $W \subseteq V$ and quantum teachers T_f for concepts f in W such that:*

1. $|W| > |V|(1 - \frac{8m^2}{l\epsilon^2})$
2. If $|\phi^{T_f}\rangle$ denotes the final superposition of L^{T_f} then, for any pair of concepts f and g of W , it holds $\| |\phi^{T_f}\rangle - |\phi^{T_g}\rangle \| < \epsilon$.

Proof. Let \mathcal{T} be the answering scheme promised by Lemma 7. We define a permutation teacher T_f for each $f \in V$ in the following way. Teacher T_f answers to query q with the answer a such that $(q, a) \in \mathcal{T}$ whenever f is consistent with (q, a) . Otherwise, any consistent basis answer is chosen in such a way that equivalent queries have equivalent answers. Note that such permutation teacher can always be constructed and it defines a valid answering scheme.

Let $|\phi_i^{\mathcal{T}}\rangle$ be the i -th query superposition of $L^{\mathcal{T}}$. Let $w(|\phi_i^{\mathcal{T}}\rangle) \in \mathbb{R}^{|Q|}$ be the $|Q|$ -dimensional vector which has entries indexed by queries $q \in Q$ and which has $w_q(|\phi_i^{\mathcal{T}}\rangle)$ as its q -th entry.

Let $w_f(|\phi_i^{\mathcal{T}}\rangle)$ be the sum of all query magnitudes $w_q(|\phi_i^{\mathcal{T}}\rangle)$ where query q is such that f is not consistent with its corresponding tuple $(q, a) \in \mathcal{T}$. Note that $w_f(|\phi_i^{\mathcal{T}}\rangle)$ is the magnitude in superposition $|\phi_i^{\mathcal{T}}\rangle$ of those queries where answering schemes T_f and \mathcal{T} are different. Moreover, observe that $Mw(|\phi_i^{\mathcal{T}}\rangle) \in \mathbb{R}^{|V|}$ is a $|V|$ -dimensional vector whose f -th entry is precisely $w_f(|\phi_i^{\mathcal{T}}\rangle)$. Since $\|M\|_1 < \frac{|V|}{l}$ and $\|w(|\phi_i^{\mathcal{T}}\rangle)\|_1 \leq 1$ we have that $\|Mw(|\phi_i^{\mathcal{T}}\rangle)\|_1 < \frac{|V|}{l}$, i. e. $\sum_{f \in V} w_f(|\phi_i^{\mathcal{T}}\rangle) < \frac{|V|}{l}$. Hence

$$\sum_{i=1}^m \sum_{f \in V} w_f(|\phi_i^{\mathcal{T}}\rangle) < \frac{m|V|}{l}. \quad (1)$$

Let us define the subset of concepts $W = \{f \in V \mid \sum_{i=1}^m w_f(|\phi_i^{\mathcal{T}}\rangle) \leq \epsilon^2/8m\}$. From (1), it follows that $|V \setminus W| < \frac{8m^2|V|}{l\epsilon^2}$. Finally, for any $f \in W$, Theorem 5 implies that $\| |\phi_m^{T_f}\rangle - |\phi_m^{\mathcal{T}}\rangle \| < \epsilon/2$. \square

Given $\epsilon = 1/8$, a non-learnability result arises from Lemma 8 whenever $|W| > 1$. Thus, it follows

Theorem 9. *Let P be a quantum protocol and let C be a target concept class. The learning query complexity of C under P holds that*

$$QC(C, P) \geq \frac{\sqrt{GHdim(C, P)}}{32}.$$

We finally note that teachers used in this discussion are permutation teachers. Thus, for popular protocols as the ones in Sect. 3.1, the statement of Theorem 9 is also valid even if only involution teachers are considered as valid oracle functions.

4.3 Upper Bounds for the Query Complexity

First in this section we provide an upper bound for deterministic learners under classical protocols in terms of the general halving dimension. This immediately yields a trivial upper bound for the quantum query complexity. Afterwards,

we show a quantum algorithm that, under many quantum protocols, achieves learning improving the trivial upper bound. Lemma 10 and Theorem 11 below can be easily proved using arguments similar to those in [5, 6]. Here, P denotes any (classical) protocol.

Lemma 10. *Let $GHdim(C, P) = d$. Then, any subset V of C with $|V| > 1$ accomplish the following predicate. There exists a query q such that for any valid answer a at least $\frac{|V|}{2d}$ concepts from V are not consistent with (q, a) .*

From Lemma 10 we get an upper bound for the query complexity.

Theorem 11. *There is a deterministic learner for the class C under protocol P whose query complexity is bounded by $\lceil 2 \ln |C| GHdim(C, P) \rceil$.*

As any quantum protocol is also a classical one and since reversible Turing machines can simulate any deterministic algorithm [8], the upper bound in Theorem 11 also applies to the quantum query complexity.

Let us consider now a quantum protocol P satisfying the following *test property*. Given a tuple (q, a) of P there is a query q' such that for any valid answer a' it is either the case that all functions consistent with (q', a') are also consistent with (q, a) or (q', a') does not share any consistent function with (q, a) . It is easy to check that protocols consisting of memberships, restricted equivalences, memberships and equivalences, and memberships and subsets hold the test property. Other protocols can also satisfy it under some specific settings. For instance, when the hypothesis class contains all singleton functions the subset protocol also holds it. On the other hand, the equivalence query protocol is one popular protocol that does not satisfy the test property.

The test property allows us to evaluate the consistency of the target function f with respect to a superposition of tuples in P by asking a query superposition. The following lemma formalizes this fact.

Lemma 12. *Let P be a quantum protocol that satisfies the test property. There is a quantum algorithm that, making a single query superposition, computes the operator that transforms the superposition of P -tuples $\sum_{p \in P} \alpha_p |p\rangle$ into $\sum_{p \in P} (-1)^{b_p} \alpha_p |p\rangle$, where $b_p = 1$ when f is not consistent with tuple p and $b_p = 0$ otherwise.*

The Grover search [14] is a nice quantum algorithm that performs a search over a space S using $O(\sqrt{|S|})$ oracle calls. We consider here an extended version of this algorithm that performs a search for a non-consistent tuple for the target f with small probability error. Lemma 13 below can be easily shown by using results in [11] and Lemma 12.

Lemma 13. *Let P be as in the previous lemma and let K be a subset of P . There is a quantum algorithm, denoted by *Extended_GS*, that provided as inputs set K and a positive integer m , makes at most $17m\sqrt{|K|}$ query superpositions, and outputs a boolean value *success* and a tuple $k \in K$ satisfying the following predicate. With error probability bounded by 2^{-m} , *success* points out if there is a non-consistent tuple in K for f and tuple k is a non-consistent one when *success* is true.*

We are ready to show a quantum learning algorithm that achieves learning under any quantum protocol holding the test property. It is inspired on previous membership queries quantum learning algorithms by Ambainis et al. [2] and Atici et al. [4]. Its query complexity will improve the trivial upper bound provided by Theorem 11 whenever GHdim is not very small.

Theorem 14. *Let P be a quantum protocol that satisfies the test property. It holds that $QC(C, P) \leq \tau \log |C| \log \log |C| \sqrt{GHdim(C, P)}$, where τ denotes a constant.*

Proof. (sketch) Let $d = GHdim(C, P)$ and let us consider the procedure Qlearner below. This procedure keeps a set of candidate functions V formed by those functions from C that have not yet been ruled out. Initially, set V agrees with C and the algorithm finishes when $|V| = 1$. We will show that at each iteration of the while loop at least $|V|/2$ functions of V are eliminated. Thus, Qlearner performs at most $\log |C|$ iterations before finishing.

procedure Qlearner (P, C)

```

1:  $V \leftarrow C$ 
2: while  $|V| \neq 1$ 
3:    $b \leftarrow \text{Is\_There\_a\_PowerfulQuery?}(P, V)$ 
4:   if  $b$  then
5:     Let  $q$  be a powerful query.
6:     Ask the basis query  $q$  and perform an observation
       on the teacher answer. Let  $a$  be the observation result.
7:      $W \leftarrow \{g \in V \mid g \text{ is not consistent with } (q, a)\}$ 
       //By the choice of  $q$ ,  $|W| \geq |V|/2$ 
8:      $V \leftarrow V \setminus W$ 
9:   else
10:    Let  $\mathcal{T}$  be an answering scheme s.t. for all  $(q, a) \in \mathcal{T}$  at least
        $|V|/2$  functions of  $V$  are consistent with  $(q, a)$ .
11:     $K \leftarrow \text{CandidateSetCover}(V, \mathcal{T})$ 
12:     $\langle \text{succes}, (q, a) \rangle \leftarrow \text{Extended\_GS}(K, \lceil \log(3 \log |C|) \rceil)$ 
13:    if succes then
14:       $W \leftarrow \{g \in V \mid g \text{ is consistent with } (q, a)\}$ 
       //By hypothesis on  $\mathcal{T}$ ,  $|W| \geq |V|/2$ 
15:       $V \leftarrow V \setminus W$ 
16:    else
17:       $W \leftarrow \{g \in V \mid \exists k \in K \text{ st } g \text{ is not consistent with } k\}$ 
       // $W$  is the subset of functions covered by  $K$ ,
       //by construction of  $K$ ,  $|W| \geq |V|/2$ 
18:       $V \leftarrow V \setminus W$ 
19:    endif
20:  endif
21: endwhile
22: return  $V$ 

```

Procedure Qlearner considers two cases in order to shrink set V . The first one — which corresponds to program lines from 5 to 8— assumes that there is a basis query q such that for any valid basis answer a at most half of the functions in V are consistent with (q, a) . Note that such q is a powerful query because asking q and making an observation on the teacher answer we can rule out at least half of the functions in V .

The second case —program lines from 10 to 19— assumes that there is no powerful query. So, for each query q there is a valid answer a such that at least half of the functions in V are consistent with (q, a) . An answering scheme \mathcal{T} formed by this type of elements is considered and a subset K of \mathcal{T} that satisfies a covering property is computed at line 11 by calling procedure CandidateSetCover below. The covering property we are interested in states that at least half of the functions in V have some non-consistency witness in K . Here, $(q, a) \in K$ is a non-consistency witness for the function $g \in V$ iff g is not consistent with (q, a) .

```

procedure CandidateSetCover( $V, \mathcal{T}$ )
   $U \leftarrow V$ 
   $K \leftarrow \emptyset$ 
  while  $|U| > |V|/2$ 
    Let  $(q, a) \in \mathcal{T}$  be such that at least  $\frac{|U|}{2d}$  concepts
    from  $U$  do not satisfy  $(q, a)$ .
    //By Lemma 10 such  $(q, a)$  always exists
     $W \leftarrow \{g \in U \mid g \text{ is not consistent with } (q, a)\}$ 
     $U \leftarrow U \setminus W$ 
     $K \leftarrow K \cup \{(q, a)\}$ 
  endwhile
  return  $K$ 
  //it holds that  $|K| \leq 2d$ 

```

By using Lemma 10, it is straightforward to show that the covering set K returned by CandidateSetCover has cardinality bounded by $2d$.

By Lemma 13, the procedure call to Extended_GS at line 12 yields, with error probability bounded by $\frac{1}{3 \log |C|}$, information about if there is a non-consistency witness in K for the target and returns a such witness if there is any. Moreover, this procedure makes at most $\tau \log \log |C| \sqrt{d}$ queries, where τ denotes a constant. Accordingly with the search success, program lines from 13 to 19 removes at least half of the functions from V .

Summarizing the results from the two cases we have considered, we conclude that, with error probability $1/3$, procedure Qlearner identifies the target concept after $\log |C|$ iterations of the while loop. \square

4.4 The General Halving Dimension and the Query Complexity of Randomized Learners

We show below that the general halving dimension also provides a lower bound for the query complexity of randomized learners under classical protocols. The results in this section are straightforward extensions of results by Simon [19]

Given a classical protocol P and a target concept class C , Simon defines a *halving game* between two deterministic players and associates a complexity to each halving game, the *halving complexity*. It can be easily shown that GHdim provides a tight characterization of this complexity. Specifically, the halving complexity is always between the value d of GHdim and $2d$. Theorem 3.1 in [19] shows a lower bound of the query complexity of randomized learners in terms of the halving complexity. This theorem immediately yields the following lower bound in terms of the general halving dimension –where the constant is different from the one in the original version because Simon defines the query complexity as an expected value–.

Theorem 15. *Any randomized learner for the target class C under protocol P with success probability $2/3$ makes at least $\frac{1}{4}GHdim(C, P)$ queries.*

5 Polynomial Learnability

We assume in this section some arbitrary underlying protocol. In order to discuss the polynomial learnability, we need to extend the concept class notion used until now. In this section a concept class C will be the union of former concept classes, i.e. $C = \cup_n C_n$ where C_n is a subset of B_n . We also need a length notion l defined on concepts in C . For instance, the length can be the circuit size. In this case, the length of concept f , denoted by $l(f)$, is the length of the minimum circuit description for function f . We assume that length notions are so that at most 2^s concepts from C have length less than s .

Given a concept class $C = \cup_n C_n$ and a length notion l , a learner L for C and l is an algorithm that accomplish the following predicate. For each n and for any target concept $f \in C_n$, given as inputs $s = l(f)$ and n and provided that a valid teacher answers the queries according to f , the algorithm L learns f . Moreover, L is a polynomial query learner when its query complexity — as a function of s and n — is bounded by a polynomial. A concept class is polynomially query learnable when it has a polynomial query learner. The following theorem, which states that any quantum polynomially learnable concept class is also polynomially learnable in the classical setting, is immediate from Theorems 9 and 11

Theorem 16. *Let C be a concept class and let $q(s, n)$ be its quantum query complexity function. Then, there exists a deterministic learner for C whose query complexity function is $O(sq^2(s, n))$.*

Under the membership query protocol Servedio and Gortler show a $O(nq^3(s, n))$ upper bound for the query complexity of deterministic learners ([17], Theorem 12). We note that this bound also follows from Theorem 16 and the $\Omega(s/n)$ lower bound for $q(s, n)$ in the membership case provided by Theorem 10 in [17].

Acknowledgments This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778, and by the spanish MCYT research project TRANGRAM, TIN2004-07925-C03-02. This publication only reflects the authors' views.

References

- [1] A. Ambainis. Quantum lower bounds by quantum arguments. *J. Comput. Syst. Sci.*, 64(4):750–767, 2002.
- [2] A. Ambainis, K. Iwama, A. Kawachi, H. Masuda, R. H. Putra, and S. Yamashita. Quantum identification of boolean oracles. In *STACS*, pages 105–116, 2004.
- [3] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.
- [4] A. Atici and R. A. Servedio. Improved bounds on quantum learning algorithms. *Quantum Information Processing*, 4(5):355–386, 2005.
- [5] J. L. Balcázar, J. Castro, and D. Guijarro. A general dimension for exact learning. In *Proceedings of the 14th Annual Conference on Computational Learning Theory*, volume 2111 of *LNAI*, pages 354–367. Springer, 2001.
- [6] J. L. Balcázar, J. Castro, and D. Guijarro. A new abstract combinatorial dimension for exact learning via queries. *J. Comput. Syst. Sci.*, 64(1):2–21, 2002.
- [7] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001.
- [8] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17:525–532, 1973.
- [9] C. H. Bennett, E. Bernstein, G. Brassard, and U. V. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997.
- [10] E. Bernstein and U. V. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997.
- [11] M. Boyer, G. Brassard, P. Hyer, and A. Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5):493–505, 1998.
- [12] 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.
- [13] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proc Roy Soc Lond A*, 439:553–558, 1992.
- [14] L. K. Grover. A fast quantum mechanical algorithm for database search. In *STOC*, pages 212–219, 1996.
- [15] L. Hellerstein, K. Pillaipakkammatt, V. Raghavan, and D. Wilkins. How many queries are needed to learn? *Journal of the ACM*, 43(5):840–862, Sept. 1996.
- [16] M. Hunziker, D. A. Meyer, J. Park, J. Pommersheim, and M. Rothstein. The geometry of quantum learning. *arXiv:quant-ph/0309059*, 2003. To appear in *Quantum Information Processing*.
- [17] R. A. Servedio and S. J. Gortler. Equivalences and separations between quantum and classical learnability. *SIAM J. Comput.*, 33(5):1067–1092, 2004.
- [18] D. R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, 1997.
- [19] H. U. Simon. How many queries are needed to learn one bit of information? *Annals of Mathematics and Artificial Intelligence*, 39:333–343, 2003.