The quantum query complexity of learning multilinear polynomials

Ashley Montanaro[*]

March 8, 2012

Abstract

In this note we study the number of quantum queries required to identify an unknown multilinear polynomial of degree d in n variables over a finite field $\mathbb{F}_q$. Any bounded-error classical algorithm for this task requires $\Omega(n^d)$ queries to the polynomial. We give an exact quantum algorithm that uses $O(n^{d-1})$ queries for constant d, which is optimal. In the case $q = 2$, this gives a quantum algorithm that uses $O(n^{d-1})$ queries to identify a codeword picked from the binary Reed-Muller code of order d.

1 Introduction

A central problem in computational learning theory is to determine the complexity of identifying an unknown function of a certain type, given access to that function via an oracle. We say that a class $\mathcal{F}$ of functions can be *learned* using t queries if any function $f \in \mathcal{F}$ can be identified with t uses of f (perhaps allowing some probability of error). It is known that some classes of functions can be learned more efficiently by quantum algorithms than is possible classically. In particular, one of the earliest results in the field of quantum computation is that the class of linear functions $\mathbb{F}_2^n \to \mathbb{F}_2$ (also known as Hadamard codewords) can be learned using a single quantum query [2], whereas $\Omega(n)$ queries are required classically. Here we generalise this result to quantum learning of *multilinear* functions over general finite fields.

Let $\mathbb{F}_q$ denote the finite field with $q = p^r$ elements for some prime p. Every function $f : \mathbb{F}_q^n \to \mathbb{F}_q$ can be represented as a polynomial in n variables over $\mathbb{F}_q$. f is said to be a degree d polynomial if it can be written as a polynomial whose every term is of total degree at most d. For example, the function $f : \mathbb{F}_5^3 \to \mathbb{F}_5$ defined by $f(x) = 2x_1 + 4x_1^2 x_2 + x_1 x_2 x_3$ is a degree 3 polynomial. The set of polynomials of degree d in n variables over $\mathbb{F}_q$ is known as the (generalised) Reed-Muller code of order d over $\mathbb{F}_q$.

We say that a degree d polynomial $f : \mathbb{F}_q^n \to \mathbb{F}_q$ is multilinear if it can be written as

$$f(x) = \sum_{S \subseteq [n], |S| \leq d} \alpha_S \prod_{i \in S} x_i$$

for some coefficients $\alpha_S \in \mathbb{F}_q$, where $[n]$ denotes the set $\{1, \ldots, n\}$. Note that in the case $S = \emptyset$ we define $\prod_{i \in S} x_i = 1$. For example, any multilinear polynomial of degree 3 can be written as

$$f(x) = \alpha_\emptyset + \sum_i \alpha_{\{i\}} x_i + \sum_{i<j} \alpha_{\{i,j\}} x_i x_j + \sum_{i<j<k} \alpha_{\{i,j,k\}} x_i x_j x_k.$$

---
[*]Centre for Quantum Information and Foundations, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK; am994@cam.ac.uk.

Technically, such functions are *multiaffine* rather than multilinear, as they are affine in each variable; however, we use the "multilinear" terminology for consistency with prior work. In particular, note that in this terminology, linear functions $f : \mathbb{F}_q^n \to \mathbb{F}_q$ (i.e. functions such that $f(x+y) = f(x)+f(y)$) are equivalent to degree 1 multilinear polynomials with no constant term. In the important special case $q = 2$ (Boolean functions), every function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is multilinear.

Given the ability to query a multilinear degree d polynomial f on arbitrary $x \in \mathbb{F}_q^n$, we would like to determine (*learn*) f using the smallest possible number of queries. A straightforward classical algorithm can solve this problem by querying $f(x)$ for all strings $x \in \mathbb{F}_q^n$ that contain only 0 and 1, and such that $|x| \leq d$. (We write $|x|$ for the Hamming weight of $x \in \mathbb{F}_q^n$, i.e. the number of non-zero components.) To see this, first consider the special case where for some k, $\alpha_S = 0$ for all S such that $|S| < k$. Then knowing $f(x)$ for all x of the above form such that $|x| = k$ is sufficient to determine all of the degree k coefficients of f (note that this relies on f being multilinear). More generally, let f_k denote the degree k part of f, i.e.

$$f_k(x) = \sum_{S \subseteq [n], |S|=k} \alpha_S \prod_{i \in S} x_i.$$

For any k, once f_ℓ is known for all $\ell \leq k$, the degree $k + 1$ coefficients can be determined from the inputs of Hamming weight $k + 1$: whenever f is queried on x, subtract $\sum_{\ell=0}^{k} f_\ell(x)$ from the result to simulate that $\alpha_S = 0$ for all S such that $|S| \leq k$. The algorithm can therefore learn f with certainty using $1 + n + \binom{n}{2} + \binom{n}{3} + \cdots + \binom{n}{d}$ queries, which is $O(n^d)$ for constant d. In the special case of functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$, all polynomials are multilinear. This implies that the class of all degree d polynomials $f : \mathbb{F}_2^n \to \mathbb{F}_2$ can be learned using $O(n^d)$ queries.

It is also easy to see that the above algorithm is exactly optimal in an information-theoretic sense. As the number of distinct multilinear degree d polynomials of n variables over $\mathbb{F}_q$ is equal to

$$q^{1+n+\binom{n}{2}+\binom{n}{3}+\cdots+\binom{n}{d}},$$

and as a classical query to f only provides $\log_2 q$ bits of information, any classical algorithm must make $1 + n + \binom{n}{2} + \binom{n}{3} + \cdots + \binom{n}{d} = \Omega(n^d)$ queries to f in order to identify it with certainty. A similar bound can be proven for bounded-error algorithms. Indeed, let f be picked uniformly at random, and consider an algorithm (without loss of generality deterministic) that makes at most c queries to f before it outputs an answer. Such an algorithm can output the correct answer for at most q^c functions f, and hence succeeds with probability at most $q^{c-(1+n+\binom{n}{2}+\binom{n}{3}+\cdots+\binom{n}{d})}$.

Using similar techniques, one can find a lower bound for *quantum* query algorithms [8]. In the standard quantum query model, the algorithm accesses f via the unitary operation $O_f |x\rangle |y\rangle = |x\rangle |y + f(x)\rangle$, where $x \in \mathbb{F}_q^n$, $y \in \mathbb{F}_q$. We formalise a lower bound on the number of queries required to identify f in this model as the following proposition[1].

Proposition 1. *Let $f : \mathbb{F}_q^n \to \mathbb{F}_q$ be a multilinear degree d polynomial over $\mathbb{F}_q$. Then any quantum query algorithm which learns f with bounded error must make $\Omega(n^{d-1})$ queries to f.*

Proof. Each query can be seen as a round of a communication process, where in each round the algorithm sends the registers $|x\rangle$ and $|y\rangle$ to the oracle, using $(n+1)\log_2 q$ qubits of communication; the oracle then performs the map $|x\rangle |y\rangle \mapsto |x\rangle |y + f(x)\rangle$ and returns the registers to the algorithm. Let f be picked uniformly at random from the set of degree d multilinear polynomials, let X

---

[1]In the case $q = 2$, this lower bound can also be obtained from independent results of Farhi et al [6] and Servedio and Gortler [13].

be the corresponding random variable, and let Y be the random variable corresponding to the function which is output by the algorithm. By Holevo's theorem [7] (see also [3]), after r rounds of communication, the mutual information between X and Y satisfies the upper bound

$$I(X:Y) \leq 2r(n+1)\log_2 q.$$

On the other hand, Fano's inequality [4] states that the probability P_e of identifying f incorrectly satisfies the lower bound

$$P_e \geq 1 - \frac{I(X:Y)+1}{\log_2\left(q^{1+n+\binom{n}{2}+\binom{n}{3}+\cdots+\binom{n}{d}}\right)},$$

which thus implies that

$$P_e \geq 1 - \frac{2r(n+1)+1/\log_2 q}{1+n+\binom{n}{2}+\binom{n}{3}+\cdots+\binom{n}{d}}.$$

For this quantity to be upper bounded by a constant, we must have $r = \Omega(n^{d-1})$. $\qquad\square$

The main result of this note is that this asymptotic scaling can actually be achieved.

Theorem 2. *Let $f : \mathbb{F}_q^n \to \mathbb{F}_q$ be a multilinear degree d polynomial over $\mathbb{F}_q$. Then there is an exact quantum algorithm which learns f with certainty using $1 + \sum_{i=1}^d 2^{i-1}\binom{n}{i-1}$ queries to f, which is $O(n^{d-1})$ for constant d.*

The case $d = 1$, $q = 2$ of this result was previously proven by Bernstein and Vazirani [2], while a bounded-error quantum algorithm using $O(n)$ queries for the case $d = 2$, $q = 2$ was more recently given by Rötteler [12]; by contrast, the algorithm given here is exact and works for all d and all fields $\mathbb{F}_q$. In related work, a quantum algorithm for estimating quadratic forms over the reals using $O(n)$ queries had previously been given by Jordan [9, Appendix D].

2   Proof of Theorem 2

The only quantum ingredient we will need to prove Theorem 2 is the following lemma, which is implicit in [1, 5] and is a simple extension of the Bernstein-Vazirani algorithm [2] for identifying linear functions over $\mathbb{F}_2$.

Lemma 3 ([1, 5]). *Let $f : \mathbb{F}_q^n \to \mathbb{F}_q$ be linear, and let $g : \mathbb{F}_q^n \to \mathbb{F}_q$ be the function $g(x) = f(x) + \beta$ for some constant $\beta \in \mathbb{F}_q$. Then f can be determined exactly using one quantum query to g.*

For completeness, we give a full proof of Lemma 3 in Appendix A.

We will derive a quantum algorithm to learn an unknown multilinear degree d polynomial f by introducing a *linear* function f_S of n variables which can be produced using a relatively small number of queries to f, and from which f can be determined using Lemma 3. This technique is somewhat similar to the approach used to learn quadratic polynomials with bounded error in the work [12]. A related function was previously used by Kaufman and Ron [10] to produce an efficient classical *tester* for low-degree polynomials over finite fields.

For any k-subset $S \subseteq [n]$, let S_j denote the j'th element of S, where S is considered as an increasing sequence of integers. For $i \in [n]$, let e_i denote the i'th element in the standard basis for

3

the vector space $\mathbb{F}_q^n$. For any $f : \mathbb{F}_q^n \to \mathbb{F}_q$ and any subset $S \subseteq [n]$, define the function $f_S : \mathbb{F}_q^n \to \mathbb{F}_q$ as follows:

$$f_S(x) = \sum_{\beta_1,\ldots,\beta_k \in \{0,1\}} (-1)^{k-\sum_{i=1}^k \beta_i} f\left(x + \sum_{j=1}^k \beta_j e_{S_j}\right),$$

where the inner sum is over $\mathbb{F}_q^n$ and the outer sum is over $\mathbb{F}_q$. For example, for $S = \{1,2\}$, $f_S(x) = f(x) - f(x + e_1) - f(x + e_2) + f(x + e_1 + e_2)$. When $q = 2$, $f_S(x)$ sums f over the affine subspace of $\mathbb{F}_2^n$ positioned at x and spanned by $\{e_i : i \in S\}$. It is clear that a query to f_S can be simulated using 2^k queries to f. One way of understanding f_S is in terms of *discrete derivative* operators. If we define the discrete derivative of f in direction $i \in [n]$ as $(\Delta_i f)(x) = f(x + e_i) - f(x)$, then $f_S(x) = (\Delta_{S_1} \Delta_{S_2} \ldots \Delta_{S_k} f)(x)$. In other words, f_S is the function obtained by taking the derivative of f with respect to all of the variables in S.

We will be interested in querying f_S for sets S of size $d - 1$. In this case, we have the following characterisation for multilinear polynomials f.

Lemma 4. *Let* $f : \mathbb{F}_q^n \to \mathbb{F}_q$ *be a multilinear polynomial of degree* d *with expansion*

$$f(x) = \sum_{T \subseteq [n], |T| \le d} \alpha_T \prod_{i \in T} x_i.$$

Then, for any S *such that* $|S| = d - 1$,

$$f_S(x) = \alpha_S + \sum_{k \notin S} \alpha_{S \cup \{k\}} x_k.$$

Lemma 4 follows easily from expressing f_S in terms of discrete derivatives; we also give a simple direct proof in Appendix B. We are now ready to describe a quantum algorithm which uses f_S to learn the degree d component of f.

Algorithm 1: Learning the degree d component of f

---

foreach $S \subseteq [n]$ *such that* $|S| = d - 1$ **do**
  | Use one query to f_S to learn the coefficients $\alpha_{S \cup \{k\}}$, for all $k \notin S$;
end
Output the function f_d defined by $f_d(x) = \sum_{S \subseteq [n], |S| = d} \alpha_S \prod_{i \in S} x_i$;

Correctness of this algorithm follows from Lemmas 3 and 4. By Lemma 4, for any S such that $|S| = d - 1$, knowledge of the degree 1 component of f_S is sufficient to determine $\alpha_{S \cup \{k\}}$ for all $k \notin S$. Therefore, knowing the degree 1 part of f_S for all $S \subseteq [n]$ such that $|S| = d - 1$ is sufficient to completely determine all degree d coefficients of f. By Lemma 3, for any S with $|S| = d - 1$, the degree 1 component of f_S can be determined with one quantum query to f_S. This implies that Algorithm 1 completely determines the degree d component of f using $\binom{n}{d-1}$ queries to f_S, each of which uses 2^{d-1} queries to f.

Once the degree d component of f has been learned, f can be reduced to a degree $d - 1$ polynomial by crossing out the degree d part whenever the oracle for f is called. That is, whenever the oracle is called on x, we subtract $f_d(x)$ from the result (recall f_d is the degree d part of f), at no extra query cost. Inductively, f can be determined completely using

$$2^{d-1}\binom{n}{d-1} + 2^{d-2}\binom{n}{d-2} + \cdots + 2n + 1 + 1$$

4

queries; the last query is to determine the constant term $\alpha_\emptyset$, which can be achieved by classically querying $f(0^n)$. The number of queries used is therefore $O(n^{d-1})$ for constant d, completing the proof of Theorem 2.

Acknowledgements

A   Quantum learning of linear functions

In order to prove Lemma 3, we will use the quantum Fourier transform (QFT) over general finite fields. This was originally defined by de Beaudrap, Cleve and Watrous [1] and independently by van Dam, Hallgren and Ip [5]. The QFT over $\mathbb{F}_q$ is defined as the unitary operation

$$Q_q|x\rangle = \frac{1}{\sqrt{q}} \sum_{y \in \mathbb{F}_q} \omega^{\mathrm{Tr}(xy)}|y\rangle,$$

where $\omega = e^{2\pi i/p}$ (recall $q = p^r$) and the trace function $\mathrm{Tr} : \mathbb{F}_q \to \mathbb{F}_p$ is defined by $\mathrm{Tr}(x) := x + x^p + x^{p^2} + \cdots + x^{p^{r-1}}$. If q is prime (i.e. $r = 1$), then of course $\mathrm{Tr}(x) = x$. The trace is linear: $\mathrm{Tr}(x+y) = \mathrm{Tr}(x) + \mathrm{Tr}(y)$ (see [11] for the proof of this and other standard facts about finite fields). This allows the n-fold tensor product of QFTs to be written concisely as

$$Q_q^{\otimes n}|x\rangle = \frac{1}{q^{n/2}} \sum_{y \in \mathbb{F}_q^n} \omega^{\mathrm{Tr}(x \cdot y)}|y\rangle,$$

where $x \cdot y = \sum_{i=1}^n x_i y_i$, the sum being taken over $\mathbb{F}_q$.

For any function $f : \mathbb{F}_q \to \mathbb{F}_q$, let U_f be the unitary operator that maps $|x\rangle \mapsto \omega^{\mathrm{Tr}(f(x))}|x\rangle$. Given access to f, U_f can be implemented using a standard phase kickback trick as follows.

Lemma 5 ([1, 5]). *U_f can be implemented using one query to f.*

Proof. To implement U_f, append an ancilla register $|y\rangle$, $y \in \mathbb{F}_q$, in the initial state $|1\rangle$. Apply Q_q^{-1} to this register to produce

$$\frac{1}{\sqrt{q}} \sum_{y \in \mathbb{F}_q} \omega^{-\mathrm{Tr}(y)}|y\rangle,$$

then apply O_f to both registers (recall $O_f|x\rangle|y\rangle = |x\rangle|y + f(x)\rangle$). For any $x \in \mathbb{F}_q$, the initial state $|x\rangle|1\rangle$ is mapped to

$$\frac{1}{\sqrt{q}}|x\rangle \sum_{y \in \mathbb{F}_q} \omega^{-\mathrm{Tr}(y)}|y + f(x)\rangle = \frac{1}{\sqrt{q}}|x\rangle \sum_{y \in \mathbb{F}_q} \omega^{-\mathrm{Tr}(y - f(x))}|y\rangle = \omega^{\mathrm{Tr}\, f(x)}|x\rangle \frac{1}{\sqrt{q}} \sum_{y \in \mathbb{F}_q} \omega^{-\mathrm{Tr}(y)}|y\rangle,$$

where we use the linearity of the trace function. As the second register is left unchanged by O_f, it can be ignored. $\qquad\square$

We are now ready to prove Lemma 3.

Lemma 3 ([1, 5]). *Let $f : \mathbb{F}_q^n \to \mathbb{F}_q$ be linear, and let $g : \mathbb{F}_q^n \to \mathbb{F}_q$ be the function $g(x) = f(x) + \beta$ for some constant $\beta \in \mathbb{F}_q$. Then f can be determined exactly using one quantum query to g.*

Proof. First observe that f will be linear if and only if $f(x) = a \cdot x = \sum_{i=1}^n a_i x_i$ for some $a \in \mathbb{F}_q^n$. Create the state

$$|\psi_g\rangle := \frac{1}{q^{n/2}} \sum_{x \in \mathbb{F}_q^n} \omega^{\operatorname{Tr}(a \cdot x + \beta)} |x\rangle$$

via the technique of Lemma 5, using one query to g. Now apply the n-fold tensor product of the inverse quantum Fourier transform to produce

$$(Q_q^{-1})^{\otimes n} |\psi_g\rangle = \frac{1}{q^n} \sum_{x \in \mathbb{F}_q^n} \omega^{\operatorname{Tr}(a \cdot x + \beta)} \sum_{y \in \mathbb{F}_q^n} \omega^{-\operatorname{Tr}(x \cdot y)} |y\rangle = \frac{1}{q^n} \omega^{\operatorname{Tr}(\beta)} \sum_{y \in \mathbb{F}_q^n} \left(\sum_{x \in \mathbb{F}_q^n} \omega^{\operatorname{Tr}((a - y) \cdot x)} \right) |y\rangle.$$

Note that β has been relegated to an unobservable global phase, and the sum over x will be zero unless $y = a$, in which case it will equal q^n. A measurement in the computational basis therefore yields a with certainty, which suffices to determine f. $\qquad\square$

B   Proof of Lemma 4

We finally prove Lemma 4, which we restate for convenience.

Lemma 4. *Let $f : \mathbb{F}_q^n \to \mathbb{F}_q$ be a multilinear polynomial of degree d with expansion*

$$f(x) = \sum_{T \subseteq [n], |T| \leq d} \alpha_T \prod_{i \in T} x_i.$$

Then, for any S such that $|S| = d - 1$,

$$f_S(x) = \alpha_S + \sum_{k \notin S} \alpha_{S \cup \{k\}} x_k.$$

Proof. For brevity, write $|\beta| = \sum_{i=1}^{d-1} \beta_i$. Let δ_{xy} be the Dirac delta function ($\delta_{xy} = 1$ if $x = y$, and $\delta_{xy} = 0$ otherwise). By the definition of f_S, for any $x \in \mathbb{F}_q^n$ we have

$$
\begin{aligned}
f_S(x) &= \sum_{\beta_1, \ldots, \beta_{d-1} \in \{0,1\}} (-1)^{d-1-|\beta|} \sum_{T \subseteq [n], |T| \leq d} \alpha_T \prod_{i \in T} \left(x_i + \sum_{j=1}^{d-1} \beta_j (e_{S_j})_i \right) \\
&= (-1)^{d-1} \sum_{T \subseteq [n], |T| \leq d} \alpha_T \sum_{\beta_1, \ldots, \beta_{d-1} \in \{0,1\}} (-1)^{|\beta|} \prod_{i \in T} \left(x_i + \sum_{j=1}^{d-1} \beta_j \delta_{S_j i} \right).
\end{aligned}
$$

Now note that for all T such that $S \not\subseteq T$, the sum over $\beta_1, \ldots, \beta_{d-1}$ will equal 0. This is because in this case there must exist an index $j \in [d-1]$ such that $S_j \notin T$, so for this j, β_j does not appear in the product over T. So, after summing over the β_i such that $i \neq j$, we are left with the sum

6

$\sum_{\beta_j \in \{0,1\}} (-1)^{\beta_j} K_T$ for some constant K_T; this evaluates to 0 for any K_T. As $|S| = d - 1$ and $|T| \le d$, this implies that we can rewrite $f_S(x)$ as

$$
\begin{aligned}
f_S(x) &= (-1)^{d-1} \alpha_S \sum_{\beta_1, \dots, \beta_{d-1} \in \{0,1\}} (-1)^{|\beta|} \prod_{i \in S} \left(x_i + \sum_{j=1}^{d-1} \beta_j \delta_{S_j i} \right) \\
&+ (-1)^{d-1} \sum_{k \notin S} \alpha_{S \cup \{k\}} \sum_{\beta_1, \dots, \beta_{d-1} \in \{0,1\}} (-1)^{|\beta|} \prod_{i \in S \cup \{k\}} \left(x_i + \sum_{j=1}^{d-1} \beta_j \delta_{S_j i} \right) \\
&= (-1)^{d-1} \alpha_S \sum_{\beta_1, \dots, \beta_{d-1} \in \{0,1\}} (-1)^{|\beta|} \prod_{i=1}^{d-1} (x_{S_i} + \beta_i) \\
&+ (-1)^{d-1} \sum_{k \notin S} \alpha_{S \cup \{k\}} \sum_{\beta_1, \dots, \beta_{d-1} \in \{0,1\}} (-1)^{|\beta|} x_k \prod_{i=1}^{d-1} (x_{S_i} + \beta_i) \\
&= (-1)^{d-1} \left(\prod_{i=1}^{d-1} \left(\sum_{\beta_i \in \{0,1\}} (-1)^{\beta_i} (x_{S_i} + \beta_i) \right) \right) \left(\alpha_S + \sum_{k \notin S} \alpha_{S \cup \{k\}} x_k \right) \\
&= \alpha_S + \sum_{k \notin S} \alpha_{S \cup \{k\}} x_k
\end{aligned}
$$

as claimed. □

References

[1] J. Niel de Beaudrap, R. Cleve, and J. Watrous. Sharp quantum versus classical query complexity separations. *Algorithmica*, 34(4):449–461, 2002. `quant-ph/0011065`.

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

[3] R. Cleve, W. van Dam, M. Nielsen, and A. Tapp. Quantum entanglement and the communication complexity of the inner product function. In *Selected papers from the First NASA International Conference on Quantum Computing and Quantum Communications*, pages 61–74, 1998. `quant-ph/9708019`.

[4] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 2006.

[5] W. van Dam, S. Hallgren, and L. Ip. Quantum algorithms for some hidden shift problems. *SIAM J. Comput.*, 36:763–778, 2006. `quant-ph/0211140`.

[6] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. How many functions can be distinguished with k quantum queries?, 1999. `quant-ph/9901012`.

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

[8] P. Høyer and R. Špalek. Lower bounds on quantum query complexity. *Bulletin of the European Association for Theoretical Computer Science*, 87:78–103, 2005. `quant-ph/0509153`.

[9] S. Jordan. *Quantum computation beyond the circuit model.* PhD thesis, MIT, 2008. `arXiv: 0809.2307`.

[10] T. Kaufman and D. Ron. Testing polynomials over general fields. *SIAM J. Comput.*, 36:779–802, 2006.

[11] R. Lidl and H. Niederreiter. *Finite fields.* Cambridge University Press, 1997.

[12] M. Rötteler. Quantum algorithms to solve the hidden shift problem for quadratics and for functions of large Gowers norm. In *Proc. MFCS'09, LNCS vol. 5734*, pages 663–674, 2009. `arXiv:0911.4724`.

[13] R. Servedio and S. Gortler. Quantum versus classical learnability. In *Proc. 16th Annual IEEE Conf. Computational Complexity*, pages 138–148, 2001. `quant-ph/0007036`.