Quantum interpolation of polynomials

Daniel M. Kane* Samuel A. Kutin†

April 2010

Abstract

Can a quantum computer efficiently interpolate polynomials? We consider black-box algorithms that seek to learn information about a polynomial f from input/output pairs $(x_i, f(x_i))$. We define a more general class of (d, S)-*independent* function properties, where, outside of a set S of exceptions, knowing d input values does not help one predict the answer. There are essentially two strategies to computing such a function: query $d+1$ random input values, or search for one of the $|S|$ exceptions. We show that, up to constant factors, we cannot beat these two approaches.

1 Introduction

We consider black-box algorithms that interpolate a polynomial from examples. But what do we mean by interpolation? It is easy to prove information-theoretic bounds on the problem of determining all coefficients of a polynomial (see Section 2). We will be interested in *function properties*, where information theory does not directly apply. A *property* of a collection $\mathcal{F}$ of functions is a (nontrivial) map $\mathcal{P}\colon \mathcal{F} \to \{0,1\}$.

Consider the set $\mathcal{F}$ of degree-d polynomials from some finite field K to itself. By EVAL we will mean a nontrivial one-bit function of $f(z)$ for some $z \in \mathcal{F}$. By COEFF we will mean a nontrivial one-bit function of some (non-constant) coefficient of f. (For example, COEFF includes asking whether a degree-d polynomial is really only degree-$(d-1)$ or less.)

Our main goal is to prove lower bounds on the number of queries needed by black-box algorithms solving EVAL or COEFF. The key observation is that both of these properties belong to the more general class of (d, S)-*independent* properties:

Definition. Let $\mathcal{F}$ be a collection of functions from some domain D to some range R. A property $\mathcal{P}\colon \mathcal{F} \to \{0,1\}$ is (d, S)-*independent* for some subset $S \subseteq D$ if, for any $z_1, \ldots, z_d \in D$ with $z_1, \ldots, z_r \in S$ and $z_{r+1}, \ldots, z_d \notin S$ (where $r = |\{z_i\} \cap S|$), the $(d-r)$-tuple of values $(f(z_{r+1}), \ldots, f(z_d))$ is independent of the $(r+1)$-tuple $(f(z_1), \ldots, f(z_r), \mathcal{P}(f))$. We say

*Dept. of Mathematics, Harvard University, One Oxford Street, Cambridge MA 02138. Email: dankane@math.harvard.edu

†IDA/CCR-P, 805 Bunn Drive, Princeton NJ 08540. Email: kutin@idaccr.org

that $\mathcal{P}$ is *d-independent* if it is $(d, \emptyset)$-independent. (For simplicity, we consider independence with respect to the uniform distribution on $\mathcal{F}$.)

Returning to the set $\mathcal{F}$ of degree-d polynomials, and allowing the domain D to be a proper subset of the field K, a one-bit function of $f(z)$ is d-independent if $z \notin D$ and $(d, \{z\})$-independent if $z \in D$. Any instance of COEFF is also d-independent. We require K to be a finite field so that knowing any d values of a polynomial gives absolutely no information about a $(d+1)$th value or about a non-constant coefficient.

We analyze quantum algorithms that compute such a $\mathcal{P}$ based on black-box access to the function f. We say that the *bias* of an algorithm is its edge over random guessing; that is, on any input f, the algorithm outputs $\mathcal{P}(f)$ with probability at least $\frac{1}{2} + \epsilon$.

We consider two models. First, in the *direct query* model, we give our oracle $x \in D$ and it returns $f(x)$. (More precisely, the quantum oracle transformation maps $|x, b, c\rangle$ to $|x, b + f(x), c\rangle$, where $+$ is some appropriate reversible notion of addition.) This is the usual black-box query model.

Theorem 1. *Let $\mathcal{P}$ be a d-independent property of a family of functions $\mathcal{F}$. Let A be a quantum query algorithm, in the direct query model, which, for any $f \in \mathcal{F}$, correctly computes $\mathcal{P}(f)$ with positive bias. Then the number of queries made by A is at least $(d+1)/2$.*

We also consider a second model, which we call the *pseudorandom PAC* model. In Valiant's classical PAC learning model [Val84], we have access to examples $(x, f(x))$, drawn according to some distribution Δ on the domain D. In the quantum PAC model, introduced by Bshouty and Jackson [BJ99], an oracle gives us the state $\sum_{x \in D} \sqrt{\Delta(x)}\, |x, f(x)\rangle$; note that measuring this state in the standard basis produces a classical PAC oracle.

In practice, however, we rarely have idealized random examples. We either have a finite list of training points, or we have a pseudorandom process that generates our examples. This can be formalized in terms of a map X from $\{1, \ldots, N\}$ to D; we give the oracle a "seed" i, and it returns $(X(i), Y(i))$, where $Y(i) = f(X(i))$. We call this the *pseudorandom PAC* model. We work with the natural quantum extension where $|i, a, b, c\rangle$ maps to $|i, a + X(i), b + Y(i), c\rangle$.

For technical reasons, we consider distributions on maps X from $\{1, \ldots, N\}$ to D. We say that such a distribution is *permutation-independent* if, for any permutation σ on D, the maps X and $\sigma \circ X$ occur with the same probability.

Theorem 2. *Let $\mathcal{P}$ be a (d, S)-independent property of a family of functions $\mathcal{F}$ with a domain of size n. Let Δ be a permutation-independent distribution of random maps. Let A be a quantum query algorithm, in the pseudorandom PAC model, which, for any $f \in \mathcal{F}$, and with $X \sim \Delta$, computes $\mathcal{P}(f)$ with bias at least ϵ. Then the number of queries made by A is at least*

$$\min\left\{\frac{d+1}{2}, C_\epsilon \sqrt{\frac{|D|}{|S|}}\right\},$$

where C_ϵ is a constant depending on ϵ.

To return to our first example, suppose $\mathcal{F}$ is the set of degree-d polynomials and $\mathcal{P}(f)$ is EVAL: one bit of information about $f(z)$ for some $z \notin D$. One strategy is to make $d+1$ queries to compute $d+1$ different values of $f(X(i))$, interpolate the polynomial, and read off $f(z)$. The above theorems show that, for either query model, this approach is within a factor of 2 of being optimal.

What if, instead, $z \in D$? In the direct query model, computing $f(z)$ is no longer interesting; we can perform a single query. In the pseudorandom PAC model, we could still query $d+1$ points and interpolate, or we could use Grover search [Gro96] to find the value of i with $X(i) = z$, at which point one additional query gives the answer. Theorem 2 says that one of these two strategies must be optimal, up to a constant factor.

We survey lower bound methods in Section 2, focusing on the approach we will use: the polynomial method [BBC$^+$]. We then prove the two above theorems in Section 3. In Section 4, we show that the bound of $(d+1)/2$ for the direct query model is tight for at least some instances of EVAL and COEFF. We conclude with some open questions in Section 5.

2 Lower Bound Methods

There are several standard techniques for proving quantum query lower bounds. One approach is to use information theory. For example, suppose our goal were not to compute a property of the degree-d polynomial f, but to produce a complete description of f. This is equivalent to specifying $d+1$ coefficients, each an element of K. But each query gives us, information-theoretically, at most two elements of K. By an interactive version of Holevo's Theorem [CvDNT99, Theorem 2], we require at least $(d+1)/4$ queries. However, this approach does not apply to computing EVAL or COEFF.

A second approach is to use the "adversary" method of Ambainis [Amb02]. The basic idea, in our setting, would be to find a collection of functions $g \in \mathcal{F}$ with $\mathcal{P}(g) = 0$, and another collection of $h \in \mathcal{F}$ with $\mathcal{P}(h) = 1$, where each g is "close to" many h, in the sense that they agree on almost all inputs. However, any two distinct polynomials disagree on almost all inputs. Høyer, Lee, and Špalek [HLŠ07], after noting that Ambainis's original method cannot prove a non-constant lower bound when 0-inputs and 1-inputs disagree on a constant fraction of the inputs, propose a variant with "negative weights" that, in theory, does not run up against this barrier. In practice, even this generalized adversary method has not yet yielded a nonconstant lower bound for such a problem.

We will apply the polynomial method [BBC$^+$]. For the direct query model, let $\delta_{x,y}$ be the function of f that is 1 when $f(x) = y$ and 0 otherwise. Then the quantum query maps

$$|x, b, c\rangle \mapsto \sum_y \delta_{x,y} |x, b+y, c\rangle .$$

So, if we start in some fixed state, after a single query each amplitude is an affine expression in the values $\delta_{x,y}$. After T queries, each amplitude is a polynomial in $\{\delta_{x,y}\}$ of degree at most T. We now measure the state and output some bit; the probability that this bit is 1 is thus a polynomial of degree at most $2T$. This polynomial p satisfies the following properties:

- If $\delta_{x,y}$ encodes any function from D to R (that is, each $\delta_{x,y} \in \{0,1\}$ and $\sum_{y \in R} \delta_{x,y} = 1$ for all x), then $0 \leq p(\{\delta_{x,y}\}) \leq 1$.
- If $\delta_{x,y}$ encodes some $f \in \mathcal{F}$, then $|p(\{\delta_{x,y}\}) - \mathcal{P}(f)| < \frac{1}{2}$.

A lower bound on the degree of such a polynomial thus gives a lower bound on the number of quantum queries.

For the pseudorandom PAC model, the same idea applies; the variable $\delta_{i,x,y}$ is 1 when $X(i) = x$ and $f(x) = y$ and 0 otherwise, and

$$|i, a, b, c\rangle \mapsto \sum_{x,y} \delta_{i,x,y} |i, a + x, b + y, c\rangle .$$

The polynomial p in this setting satisfies the properties:

- If $\delta_{i,x,y}$ encodes any functions X from indices to D and Y from indices to R (that is, each $\delta_{i,x,y} \in \{0,1\}$ and $\sum_{x,y} \delta_{i,x,y} = 1$ for all i), then $0 \leq p(\{\delta_{i,x,y}\}) \leq 1$. (Even if X and Y do not encode a function, the algorithm still outputs 1 with some probability.)
- If $\delta_{i,x,y}$ encodes X and $f \circ X$ for some $f \in \mathcal{F}$, then $\mathop{\mathbf{E}}_{X \sim \Delta} \left[|p(\{\delta_{i,x,y}\}) - \mathcal{P}(f)| \right] \leq \frac{1}{2} - \epsilon$.

In early uses of the polynomial method [BBC+], one step in a typical application was to symmetrize down to a polynomial in one variable. This works well for total functions, but not for promise problems. (Here, the promise is that f represent some function.) The method has been adapted to a similar setting for proving a lower bound for the element distinctness problem [AS04, Kut05]; in this case, symmetrizing separately on the domain and range yields a function of two variables. We will use a similar approach to tackle interpolation.

Remark. There are different ways to prove lower bounds on the degree of a polynomial computing a function. For example, a referee for an early version of this paper noted that Theorem 1 above can be proved using a general result[1] of Buhrman, et al. [BVdW07]. We give a direct proof whose main idea generalizes to the pseudorandom PAC model.

3 Proofs

We now prove our main results. We begin with the direct query model.

Proof of Theorem 1. Let A be an algorithm computing the d-independent property $\mathcal{P}$ with nonzero bias. Suppose that A makes fewer than $(d+1)/2$ queries. As discussed in Section 2, we write the probability that A outputs 1 as a polynomial $p(f)$, by which we mean a polynomial in the variables $\{\delta_{x,y}\}$, of degree at most d. When $f \in \mathcal{P}^{-1}(0)$, then $0 \leq p(f) < \frac{1}{2}$; when $f \in \mathcal{P}^{-1}(1)$, then $\frac{1}{2} < p(f) \leq 1$.

[1] See the discussion following their Lemma 3 [BVdW07].

Write p as a sum of monomials $\sum_k m_k$. Each monomial has the form

$$m_k = \prod_{j=1}^{t} \delta_{x_j,y_j}$$

for some $t \leq d$. Hence, each m_k depends on at most d values of f. By the definition of d-independence, the expected value of m_k over $\mathcal{P}^{-1}(0)$ is the same as it is over $\mathcal{P}^{-1}(1)$. This is true for all k, so

$$\frac{1}{2} < \mathop{\mathbf{E}}_{f \in \mathcal{P}^{-1}(1)}[p(f)] = \mathop{\mathbf{E}}_{f \in \mathcal{P}^{-1}(0)}[p(f)] < \frac{1}{2}.$$

This is impossible. We conclude that no such algorithm exists; that is, any algorithm computing $\mathcal{P}$ requires at least $(d+1)/2$ queries. □

The proof of Theorem 2 is more involved. We will first show that, assuming an algorithm makes fewer than $(d+1)/2$ queries, the actual values of $f(x)$ do not matter unless x is in the special set S. This first part of the argument uses the same logic as the proof of Theorem 1.

Intuitively, if the values $f(x)$ matter only for $x \in S$, the simplest possible case would be one where any such value of $f(x)$ immediately yields the answer $\mathcal{P}(f)$. This is Grover search, with a known lower bound of $\Omega(\sqrt{|D|/|S|})$. The second part of the proof of Theorem 2 represents one approach to formalizing this intuition.

Proof of Theorem 2. Let A be an algorithm computing the (d, S)-independent property $\mathcal{P}$ with bias at least ϵ. Suppose that A makes fewer than $(d+1)/2$ queries. As discussed in Section 2, we write the probability that A outputs 1 as a polynomial $p(X, Y)$, by which we mean a polynomial in the variables $\{\delta_{i,x,y}\}$, of degree at most d. For any i and any $x \notin S$, we introduce the variables $\xi_{i,x}$ (which is 1 when $X(i) = x$ and 0 otherwise) and $\upsilon_{i,y}$ (which is 1 when $Y(i) = y$ and 0 otherwise), and we write $\delta_{i,x,y} = \xi_{i,x}\upsilon_{i,y}$. For all $X\colon \{1, \ldots, n\} \to D$ and $Y\colon \{1, \ldots, n\} \to R$, we have $0 \leq p(X, Y) \leq 1$; we will use this generality. When $f \in \mathcal{F}$, we have $|p(X, f \circ X) - \mathcal{P}(f)| \leq \frac{1}{2} - \epsilon$.

Write p as a sum of monomials $\sum_k m_k$. Each monomial has the form

$$m_k = \prod_{j=1}^{r} \delta_{i_j,x_j,y_j} \prod_{j=r+1}^{t} \xi_{i_j,x_j}\upsilon_{i_j,y_j}$$

for some $r \leq t \leq d$ with $x_j \in S$ for $j \leq r$ and $x_j \notin S$ for $j > r$, and with all i_j distinct. By the definition of d-independence, once we condition on $X(i_j) = x_j$ for $1 \leq j \leq t$, the expected value of $\prod_{j=r+1}^{t} \upsilon_{i_j,y_j}$ over $f \in \mathcal{F}$ and $X \sim \Delta$ is independent of $\mathcal{P}(f)$ and of the values δ_{i_j,x_j,y_j} for $j \leq r$. Hence, we can replace this product with its expected value over f and X, yielding a new polynomial q using only the variables $\delta_{i,x,y}$ (for $x \in S$) and $\xi_{i,x}$ (for $x \notin S$). The polynomial $q(X, Y)$ can be viewed as an average over p, so it satisfies the original conditions: $0 \leq q(X, Y) \leq 1$ for any X, Y, and $\mathop{\mathbf{E}}_{X \sim \Delta}\left[|q(X, f \circ X) - \mathcal{P}(f)|\right] \leq \frac{1}{2} - \epsilon$ when $f \in \mathcal{F}$. Furthermore, $\deg q \leq \deg p$.

If $S = \emptyset$, then q depends only on X but not f, which is impossible. In this case, A must have made at least $(d+1)/2$ queries. For the remainder of the proof we assume S is nonempty.

We now apply q to a particular set of instances. Let $n = |D|$ and $k = |S|$. Write $S = \{z_1, \ldots, z_k\}$, and write $D \setminus S = \{z_{k+1}, \ldots, z_n\}$. We will permute these values in blocks. Let $B = \lfloor n/k \rfloor$. For any function π from $\{0, \ldots, B-1\}$ to $\{0, \ldots, B-1\}$ we get a function σ on D given by $\sigma(z_{a+kb}) = z_{a+k\pi(b)}$ for $1 \le a \le k$ and $0 \le b < B$. (We write $\sigma(z_a) = z_a$ for $a > Bk$.) If π is a permutation, then so is σ.

Now, fix some $g, h \in \mathcal{F}$ with $\mathcal{P}(g) = 0$ and $\mathcal{P}(h) = 1$. We describe a new distribution on maps $X \colon \{1, \ldots, N\} \to D$ and $Y \colon \{1, \ldots, N\} \to R$. First, choose $X_0 \sim \Delta$ and a uniform random permutation π on $\{0, \ldots B-1\}$. Construct σ from π as above. Let $X = \sigma \circ X_0$. If $X_0(i) = z_{a+kb}$, with $1 \le a \le k$ and $0 \le b < B$, then let $Y(i)$ be $g(z_a)$ when b is even and $h(z_a)$ when b is odd.

We now consider $Q(X_0, \pi) = q(\sigma \circ X_0, Y)$, with σ and Y as above. We can express each $\delta_{i,x,y}$ or $\xi_{i,x}$ as a bilinear combination of variables $\theta_{i,x}$ (1 if $X_0(i) = x$ and 0 otherwise) and $\eta_{i,j}$ (1 if $\pi(i) = j$ and 0 otherwise). Hence, $Q(X_0, \pi)$ is a polynomial in $\{\theta_{i,x}\}$ and $\{\eta_{i,j}\}$.

For any X_0 and function π, we must have $0 \le Q(X_0, \pi) \le 1$. Let $\bar{Q}(\pi) = \mathop{\mathbf{E}}_{X_0 \sim \Delta} Q(X_0, \pi)$. For a permutation π with $\pi^{-1}(0)$ even, we have $\bar{Q}(\pi) = \mathop{\mathbf{E}}_{X \sim \Delta}[q(X, g \circ X)] \le \frac{1}{2} - \epsilon$. This is where we use the fact that Δ is permutation-independent, which implies that choosing $X_0 \sim \Delta$ and $X \sim \Delta$ are equivalent. Similarly, for a permutation π with $\pi^{-1}(0)$ odd, we have $\bar{Q}(X_0, \pi) = \mathop{\mathbf{E}}_{X \sim \Delta}[q(X, h \circ X)] \ge \frac{1}{2} + \epsilon$. Note that $\bar{Q}$ is a polynomial of degree at most d in $\{\eta_{i,j}\}$.

We have reduced to the standard problem of permutation inversion; as first shown by Ambainis [Amb02], we know that any such polynomial has degree $\Omega(\sqrt{B})$. For concreteness, we finish the proof using symmetrization. First, we symmetrize $\bar{Q}$ with respect to any rearrangement of the values $1, \ldots, B-1$ in the range of π. This reduces us to variables $\{\eta_i\}$ where $\eta_i = 1$ when $\pi(i) = 0$ and 0 otherwise. Next, we symmetrize with respect to any rearrangement of even i and any rearrangement of odd i. We are left with a polynomial $P(\alpha, \beta)$ in two variables: α counts the number of even i with $\pi(i) = 0$, and β counts the number of odd i with $\pi(i) = 0$.

Note that $0 \le P(\alpha, \beta) \le 1$ for any $0 \le \alpha \le \lceil B/2 \rceil$ and any $0 \le \beta \le \lfloor B/2 \rfloor$. Furthermore, $P(1, 0) \le \frac{1}{2} - \epsilon$ and $P(0, 1) \ge \frac{1}{2} + \epsilon$. We break into two cases depending on whether $P(0, 0)$ is at least $\frac{1}{2}$ or at most $\frac{1}{2}$. In either case, we get a polynomial $\hat{P}$ in one variable with $0 \le \hat{P}(i) \le 1$ for $i = 0, \ldots, \lfloor B/2 \rfloor$ and with a constant gap between $\hat{P}(0)$ and $\hat{P}(1)$. By a lemma of Paturi [Pat92] (see also [BBC$^+$, NS94]), we conclude that $\deg \hat{P} = \Omega(\sqrt{B})$ as desired. By construction, $\deg \hat{P} \le \deg p$. $\square$

4 Characteristic 2

We now return to the direct query model. We have proven a lower bound of $(d+1)/2$ queries for EVAL or COEFF. The usual classical interpolation algorithm, of course, requires $d+1$ queries. We suspect that, in full generality, a quantum algorithm for EVAL or COEFF would also require $d+1$ queries, although we do not have a proof. We now show that we can solve some instances of EVAL or COEFF in only $(d+1)/2$ queries. In these cases, we have an exact bound.

Before we describe these instances, we give a simple argument that, in the generality in which it is stated, Theorem 1 is tight. Let $\mathcal{F}$ be the set of all functions from some domain to $\{0,1\}$, and let $\mathcal{P}(f)$ be the parity $\bigoplus_{x\in V} f(x)$ of some collection of input places with $|V| = d+1$. This is a d-independent property; any set of d values, even if they all lie in V, are independent of the final answer. In this case, the standard Deutsch–Josza algorithm [DJ92] computes the parity with $(d+1)/2$ queries.[2]

Lemma 3. *Let $K = \mathbb{F}_{2^k}$. Let $V \subseteq K$ be an affine $\mathbb{F}_2$-subspace of K of dimension $\ell \leq k$, and let x_0 denote one element of V. For any $t \leq 2^\ell - 2$, we have*

$$\sum_{x\in V} x^t = 0.$$

For $t = 2^\ell - 1$, we have

$$\sum_{x\in V} x^t = \prod_{x\in V\setminus\{x_0\}} (x - x_0).$$

Proof. There are various ways to prove this result. (See, for example, Van Lint [Lin98, Lemma 6.6.12].) One approach is to choose a basis $v_1, \ldots, v_\ell$ for $V + x_0$, so that each $x \in V$ is of the form $x_0 + \sum \lambda_i v_i$ with $\lambda_i \in \{0,1\}$. Now, for $t \leq 2^\ell - 1$, write out $\sum_{x\in V} x^t$ as a degree-t polynomial p in the variables x_0 and v_i.

If we evaluate p at any input where the v_i are linearly dependent, then we are in fact summing x^t over a subspace of dimension $\ell' < \ell$, in which case each term occurs $2^{\ell-\ell'}$ times, and $p(x_0, v_1, \ldots, v_\ell) = 0$. We conclude that, for each nonzero choice of $\{\lambda_i\}$, $\sum \lambda_i v_i$ divides p, and hence $q = \prod_{\lambda_i \text{ not all } 0}(\sum \lambda_i v_i)$ is a factor of p.

Note that $\deg q = 2^\ell - 1$. So, if $t \leq 2^\ell - 2$, $q \mid p$ implies $p = 0$. For $t = 2^\ell - 1$, we must have $p = Cq$ for some constant C; checking coefficients reveals $C = 1$. □

We now show how to hide a parity problem inside EVAL. Choose positive integers ℓ and k with $2 \leq \ell \leq k$; we take K to be the field $\mathbb{F}_{2^k}$, and we take $d = 2^\ell - 2$. Let $V \subseteq K$ be an affine $\mathbb{F}_2$-subspace of dimension ℓ. Choose $x_0 \in V$, and let $X = V \setminus \{x_0\}$. Finally, let $L\colon K \to \mathbb{F}_2$ be some $\mathbb{F}_2$-linear functional on K. We consider the EVAL problem of computing $L(f(x_0))$ by querying[3] $f(x)$ for $x \in X$.

[2] In fact, Theorem 1 can be viewed as an extension of the parity lower bound of Farhi, et al. [FGGS98].

[3] Here, when we apply our query $|x, b, c\rangle \mapsto |x, b + f(x), c\rangle$, we require that "+" denotes addition in the field K (i.e., bitwise addition).

Proposition 4. *For K, d, f, L, x_0, X as above, we can compute $L(f(x_0))$, with no error, in $\lceil (d+1)/2 \rceil$ queries.*

Proof. By Lemma 3, $\sum_{x \in V} x^t = 0$ for any $t \leq d$, and hence $\sum_{x \in V} f(x) = 0$. Equivalently, we have $f(x_0) = \sum_{x \in X} f(x)$, and hence $L(f(x_0)) = \sum_{x \in X} L(f(x))$.

We now note that we can modify our query transformation to output $L(f(x))$ rather than $f(x)$. If $f(x)$ is the k-bit quantity $y_1 \ldots y_k$, then L must be of the form $\bigoplus_{i \in S} y_i$ for some subset S of bits. Let ϕ be the state in the output register where the bits in S are in state $(|0\rangle - |1\rangle)/\sqrt{2}$, and the bits not in S into $(|0\rangle + |1\rangle)/\sqrt{2}$. Then our query maps $|x, \phi, z\rangle$ to $(-1)^{L(f(x))} |x, \phi, z\rangle$.

Now, using the Deutsch–Jozsa technique [DJ92], we need only $\lceil |X|/2 \rceil$ queries to compute $L(f(x_0))$, as desired. □

A simliar approach works for COEFF. As before, choose $2 \leq \ell \leq k$ and take $K = \mathbb{F}_{2^k}$. Now, take $d = 2^\ell - 1$. Write $f(x) = \sum_{i=0}^{d} a_i x^i$; we let $L \colon K \to \mathbb{F}_2$ be some $\mathbb{F}_2$-linear functional on K, and we ask for the value of $L(a_d)$. We let $X \subseteq K$ be an affine subspace of dimension ℓ.

Proposition 5. *For K, d, f, L, X as above, we can determine $L(a_d)$ in $(d+1)/2$ queries.*

Proof. Choose $x_0 \in X$. By Lemma 3,

$$\sum_{x \in X} f(x) = a_d \prod_{x \in X \setminus \{x_0\}} x.$$

The right-hand side is simply $a_d c$, where c is a nonzero constant depending only on X. Let μ denote multiplication by c; this is an $\mathbb{F}_2$-linear map. Hence, $\bigoplus_{x \in X} (L \circ \mu^{-1})(f(x)) = L(a_d)$. As in Proposition 4, we can query $\bigoplus_{x \in X} (L \circ \mu^{-1})(f(x))$ using only $|X|/2$ queries. □

5 Conclusions

We have proven a lower bound of $(d+1)/2$ for polynomial interpolation, whether we consider interpolation to be evaluating at a point (outside our query set) or computing a coefficient. We showed in Section 4 that, in some cases in characteristic 2, this lower bound is tight. It would be interesting to prove a lower bound of $d+1$ for other characteristics, or to give a family of examples in another characteristic where we require only αd queries for some constant $\alpha < 1$.

We also introduce a pseudorandom PAC model, and show that, in this model, either interpolation or Grover's search gives an optimal solution to the EVAL problem. We conjecture that the lower bound of Theorem 2 also applies to the Bshouty–Jackson quantum PAC model.

Finally, we hope that the concept of d-independent function properties will prove to be useful for other quatum lower bound problems.

Acknowledgments

The authors thank Bruce Jordan for proposing this problem. We also thank Bob Beals, David Moulton, and Mike Zieve for helpful discussions.

References

[Amb02] Andris Ambainis. Quantum lower bounds by quantum arguments. *J. Comp. Sys. Sci.*, 64:750–767, 2002.

[AS04] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004.

[BBC+] Robert Beals, Howard Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. quant-ph/9802049. IEEE Symposium on Foundations of Computer Science '98.

[BJ99] Nader H. Bshouty and Jeffrey C. Jackson. Learning DNF over the uniform distribution using a quantum example oracle. *SIAM J. Comput.*, 28(3):1136–1153, 1999.

[BVdW07] Harry Buhrman, Nikolay Vereshchagin, and Ronald de Wolf. On computation and communication with small bias. In *Proceedings of the 22nd annual IEEE Conference on Computational Complexity (CCC)*, 2007.

[CvDNT99] Richard Cleve, Wim van Dam, Michael Nielsen, and Alain Tapp. Quantum entanglement and the communication complexity of the inner product function. In *Quantum Computing and Quantum Communications, First NASA International Conference, QCQC '98*, number 1509 in Lecture Notes in Computer Science, pages 61–74. Springer, 1999.

[DJ92] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London*, A439:553–558, 1992.

[FGGS98] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Limit on the speed of quantum computation in determining parity. *Phys. Rev. Lett.*, 81(24):5442–5444, Dec 1998.

[Gro96] Lov Grover. A fast quantum mechanical algorithm for database search. ACM Symposium on Theory of Computing (updated version), 1996. quant-ph/9605043.

[HLŠ07] Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Symposium on Theory of Computing*, STOC '07, 39th Annual ACM Symposium on Theory of Computing, pages 526–535, June 2007.

[Kut05] Samuel Kutin. Quantum lower bound for the collision problem with small range. *Theory of Computing*, 1:29–36, 2005.

[Lin98] J.H. van Lint. *Introduction to Coding Theory*. Springer, third edition, 1998.

[NS94] Noam Nisan and Mario Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, 1994.

[Pat92] Ramamohan Paturi. On the degree of polynomials that approximate symmetric boolean functions. In *Symposium on Theory of Computing*, STOC '92, 24th Annual ACM Symposium on Theory of Computing, pages 468–474, 1992.

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