comput. complex. **24** (2015), 255 – 293

1016-3328/15/020255-39
published online April 17, 2015
DOI 10.1007/s00037-015-0099-2

computational complexity

QUANTUM ALGORITHMS FOR LEARNING SYMMETRIC JUNTAS VIA THE ADVERSARY BOUND

Aleksandrs Belovs

Abstract. In this paper, we study the following variant of the junta learning problem. We are given oracle access to a Boolean function f on n variables that only depends on k variables, and, when restricted to them, equals some predefined function h. The task is to identify the variables the function depends on. When h is the XOR or the OR function, this gives a restricted variant of the Bernstein–Vazirani or the combinatorial group testing problem, respectively.
We analyze the general case using the adversary bound and give an alternative formulation for the quantum query complexity of this problem. We construct optimal quantum query algorithms for the cases when h is the OR function (complexity is $\Theta(\sqrt{k})$) or the exact-half function (complexity is $\Theta(k^{1/4})$). The first algorithm resolves an open problem from Ambainis & Montanaro (Quantum Inf Comput 14(5&6): 439–453, 2014). For the case when h is the majority function, we prove an upper bound of $O(k^{1/4})$. All these algorithms can be made exact. We obtain a quartic improvement when compared to the randomized complexity (if h is the exact-half or the majority function), and a quadratic one when compared to the non-adaptive quantum complexity (for all functions considered in the paper).

Keywords. Quantum query algorithms, computational learning theory, combinatorial group testing, representation theory of the symmetric group, semi-definite optimization.

Subject classification. 68Q12, 81P68.

Birkhäuser

1. Introduction

Learning theory studies the problem of reconstructing functions from their values in various points. In this paper, we study the problem of exact learning from membership queries. In this problem, one is given oracle access to a function $f\colon \{0,1\}^n \to \{0,1\}$ belonging to some fixed class of functions $\mathcal{C}$ (usually called *concept class*). The task is to identify the function using the smallest possible number of queries to the oracle. It is required to give the exact description of the function, not an approximation (although, it is allowed to err with small probability like 1/3).

This is a broad area of research both classically and quantumly. We shall highlight some of the results. Classically, the problem was defined by Angluin (1988). Bshouty *et al.* (1996) obtained upper and lower bounds on the randomized query complexity of learning a concept class $\mathcal{C}$ exactly using a combinatorial parameter $0 < \hat{\gamma}^{\mathcal{C}} \leq 1$ of the class. More specifically, the query complexity is $O\big(\frac{\log|\mathcal{C}|}{\hat{\gamma}^{\mathcal{C}}}\big)$ and $\Omega\big(\frac{1}{\hat{\gamma}^{\mathcal{C}}} + \log|\mathcal{C}|\big)$.

Quantumly, this problem was analyzed (under the name of quantum oracle interrogation or identification) by van Dam (1998) and Ambainis *et al.* (2004). Van Dam considered the case when $\mathcal{C}$ consists of all Boolean functions on n variables, where $n/2+O(\sqrt{n})$ quantum queries suffice, in contrast to n queries required classically. Ambainis *et al.* constructed a quantum algorithm for the general case with query complexity $O(\sqrt{n\log|\mathcal{C}|\log n}\log\log|\mathcal{C}|)$. Finally, Kothari (2014) gave a complete characterization of the quantum query complexity of this problem in terms of n and $|\mathcal{C}|$.

Servedio & Gortler (2004) proved some quantum analogues of the results in Bshouty *et al.* (1996). In particular, they showed that, for any concept class $\mathcal{C}$, the quantum query complexity of learning $\mathcal{C}$ exactly is $\Omega\Big(\frac{1}{\sqrt{\hat{\gamma}^{\mathcal{C}}}} + \frac{\log|\mathcal{C}|}{n}\Big)$. Using this result, they obtained that the deterministic complexity of the same problem is $O(nQ^3)$ where Q is its quantum query complexity. Atıcı & Servedio (2005) constructed a quantum $O\Big(\frac{\log|\mathcal{C}|\log\log|\mathcal{C}|}{\sqrt{\hat{\gamma}^{\mathcal{C}}}}\Big)$-query algorithm for the same problem.

The problem and related work In this paper, we study the following learning problem proposed by Ambainis & Montanaro (2014). Let $h\colon \{0,1\}^k \to \{0,1\}$ be a fixed symmetric Boolean function. We are given oracle access to a Boolean function f on $n \gg k$ variables that satisfies the following properties. The function f only depends on a subset A of k input variables, and, when restricted to these variables, the function equals h. Thus, the learning problem reduces to identifying the set A.

Functions that only depend on a small number of the input variables are called *juntas*. Thus, our problem is related to the problem of learning and testing juntas, which has been studied both classically (see Blais 2009 and the references therein) and quantumly (Atıcı & Servedio 2007). Note, however, that our settings are different from that of usual junta learning. First, we have an additional promise that the function f equals the function h. Second, we are allowed adaptive membership queries, not only samples. And third, we have to find the function f exactly, not an approximation. The last two aspects make our settings different from the quantum PAC model (Bshouty & Jackson 1998).

From a simple information-theoretical argument it follows that $\Omega(\log|\mathcal{C}|) = \Omega(k \log \frac{n}{k})$ randomized queries are required to solve this problem classically. Quantumly, as usual, one can do better. One of the pioneering quantum algorithms, the Bernstein-Vazirani algorithm (Bernstein & Vazirani 1997), can be stated in these settings. The algorithm solves our problem for the case when h is the XOR function. It does so in one query, without an error, and, moreover, for all values of k simultaneously.

Another example is the combinatorial group testing problem (despite the name, it is a *learning* problem). In this problem, a set X of n elements is given, and it is known that at most k of them are marked. For any subset $S \subseteq X$, it is possible to detect, in one query, whether S contains a marked element. The task is to identify all marked elements making as few queries as possible. It corresponds to the case when h is the OR function (if we additionally require having *exactly* k marked elements). This is a well-studied problem classically (Du & Hwang 1993). Ambainis & Montanaro (2014) studied the quantum complexity of this problem

and its special case, search with wildcards, that we do not define here. The search with wildcards problem was resolved, but the complexity of the combinatorial group testing problem was only stated to lie between $\Omega(\sqrt{k})$ and $O(k)$.

The quantum counterfeit coin problem studied by Iwama *et al.* (2012) is also closely connected to our work. In this problem, one is given n coins, and it is known that exactly k of them are counterfeit. All genuine coins have the same weight, all counterfeit coins have the same weight, and the counterfeit coins are strictly lighter than the genuine ones. One is also given perfect scales, and the task is to find all counterfeit coins using as few weighing operations as possible. More formally, the oracle accepts two disjoint equal-sized subsets $S, T \subseteq [n]$ as its input. It replies with 0 if S and T contain equal number of counterfeit coins, and with 1 otherwise. (I.e., one only gets to know whether the scales are balanced or not.) Iwama *et al.* constructed a quantum algorithm that solves this problem in $O(k^{1/4})$ queries to the oracle. No general lower bound is known for this problem.

Our contribution In this paper, we do the following. In Section 3, we resolve the question posed by Ambainis and Montanaro by describing a tight quantum $O(\sqrt{k})$-query algorithm for the combinatorial group testing problem (in its full generality, i.e., allowing less than k marked elements). In Section 4, we use the adversary bound and representation theory to formulate an optimization problem for the quantum query complexity of our learning problem for any symmetric function h. In Section 5, we solve this optimization problem when h is the exact-half function (the function that evaluates to 1 iff exactly $\lfloor k/2 \rfloor$ of the input variables equal 1). The quantum query complexity of the learning problem turns out to be $\Theta(k^{1/4})$. In Section 6, we describe some partial results for the case when h is the majority function. Finally, in Section 7, we show that most of the above algorithms can be made exact without increase in their complexity, and prove some no-go results for non-adaptive quantum algorithms.

Previous techniques Before discussing our techniques, let us describe some previously used techniques. One possibility is to

apply the Grover search (as in the papers by Ambainis *et al.* 2004, and Atıcı & Servedio 2005). This gives at most quadratic speed-up.

Most of the papers, however, use the following *prepare-and-measure* strategy: A quantum state $|\psi\rangle$ is prepared, a tensor power $O_x^{\otimes T}$ of the input oracle is applied to the state, and the result is measured. This strategy usually comes in one of the two variations. The first one is *Fourier sampling*. In this case, $T = 1$ and $|\psi\rangle$ is the uniform superposition. The resulting state, $O_x|\psi\rangle$, is measured in the Fourier basis. This procedure is repeated many times, and when enough samples have been collected, they are processed by a classical subroutine to reconstruct f. Notable examples are the DNF learning algorithm by Bshouty & Jackson (1998) and the junta learning algorithm by Atıcı & Servedio (2007), where this approach is mentioned explicitly (under the name of quantum example oracle in the first paper, and Fourier sampling oracle in the second one).

A more general variant is to show that the states $O_x^{\otimes T}|\psi\rangle$ and $O_y^{\otimes T}|\psi\rangle$ are almost orthogonal for all $x \neq y$ and then apply the Pretty Good Measurement (Hausladen & Wootters 1994) to distinguish them. Examples here are Childs *et al.* (2013); Ettinger *et al.* (2004).

Either way, the prepare-and-measure strategy usually can be made non-adaptive (see Section 2 for the definition). This is a limitation. For example, Zalka (1999) showed that a non-adaptive quantum algorithm requires $\Omega(n)$ queries to solve the OR function, in contrast to the Grover search. Childs *et al.* (2013) explain why their hidden shift algorithm performs sub-optimally on the delta function using this argument.

All these approaches are unsatisfactory for our problem. First, the general results mentioned in the beginning of this section are useless here, because the quantum query complexity of our problems is less than k, which is much less than n or $\log|\mathcal{C}|$. Next, we attain super-quadratic speed-ups over randomized algorithms, which is not possible by only using the Grover search. Finally, in Section 7, we show that any *non-adaptive* quantum algorithm requires quadratically more queries than our algorithms. This does not completely rule out the prepare-and-measure strategy,

but shows that its easiest and most common one-shot variant does not work here.

It is also interesting to compare our algorithm for the exactly-half function to the algorithm for the counterfeit coin problem by Iwama *et al.* (2012). After all, both algorithms attain complexity $O(k^{1/4})$, which is a quartic improvement to the randomized complexity. We are not aware of any reduction in either of two directions. Iwama *et al.* reduce the counterfeit problem to the Bernstein-Vazirani problem. Indeed, if an even-sized subset S contains even number of counterfeit coins, there exist dissections of S into two equal-sized subsets having equal number of counterfeit coins. These dissections can be detected using quantum amplitude amplification (Brassard *et al.* 2002). It seems unlikely that a similar approach can be applied for the exact-half function.

Our techniques Instead of these techniques, we use the dual adversary bound. The adversary bound is a lower bound on quantum query complexity first developed by Ambainis (2002) in the form that is now known as the positive-weighted adversary. Later, it was strengthened by Høyer *et al.* (2007) to the negative-weighted, or general adversary bound. Reichardt *et al.* proved that this lower bound is tight by showing how the dual to the adversary bound can be converted into a quantum query algorithm (Lee *et al.* 2011; Reichardt 2009). Their algorithm is based on quantum walks.

Thus, a quantum query algorithm can be constructed by coming up with a feasible solution to the dual adversary bound. There has been some work in this vein. One example is provided by algorithms for formulae evaluation (Reichardt & Špalek 2012; Zhan *et al.* 2012). Another line of development is learning graphs (Belovs 2012b). They were applied to improve quantum query complexity of triangle and other subgraph detection (Belovs & Reichardt 2012; Lee *et al.* 2013), and the k-distinctness problem (Belovs 2012a). In general, learning graphs work well for Boolean functions with small 1-certificates. Clearly, both of these general approaches do not work here. Indeed, our problem does not have a nice formula description, nor does it have Boolean output, nor small certificates.

Instead of that, we construct a feasible solution to the dual adversary from scratch. Let us give a short overview of our construction. For precise formulations of the adversary bound, the reader may refer to Section 2. Informally, the dual adversary bound (2.4) boils down to distinguishing inputs $A, B \in \mathcal{C}$ using queries (2.4b). In the following informal exposition, we analyze complexity of distinguishing A and B using both a usual randomized algorithm and the adversary bound, and compare the two. Although obtaining equality in (2.4b), and not a lower bound like in (2.6), is important, we ignore this issue for now.

We start with combinatorial group testing, which corresponds to the case when h is the OR function. Assume we want to distinguish k-subsets $A, B \subseteq [n]$. Moreover, we want to do so regardless of the distance $\ell = |B \setminus A|$. A simple strategy is to take a subset $S \subseteq [n]$ by including each element of $[n]$ with probability p independently at random, and hope that exactly one of $S \cap A$ and $S \cap B$ is empty.

Classically, the worst case is when the distance $\ell = 1$. In this case, conditioned on $S \cap A = \emptyset$, the probability that S distinguishes A and B (i.e., that $S \cap B \neq \emptyset$) is p. But taking $p \gg 1/k$ does not make much sense, because then the probability that S does not intersect A is too small.

The dual adversary, however, allows for additional tricks. In particular, we may "condition" on S and A having intersection of size at most 1. That is, the queries S with $|S \cap A| \geq 1$ count neither toward the complexity, nor toward distinguishing A and B. (The same, clearly, applies for B as well.) Thus, in this settings, we may even take $p = 1/2$, which increases the chances of A and B being distinguished.

But when ℓ is, say, k, the choice of $p = 1/2$ does not work. Indeed, conditioned on $A \cap S = \emptyset$, the probability of $|S \cap B| = 1$ is very small. (Remember, we do not use S for B if $|S \cap B| > 1$.) In this case, $p = 1/k$ is a much better choice. In the final solution, we take $p \in (0, 1)$ uniformly at random that, a bit surprisingly, works for all values of ℓ.

Thus, our solution to the combinatorial group testing problem is somewhat *ad hoc*. The analysis is so simple because we may

assume that S intersects A in either 0 or 1 element. If h is the majority function, it is suboptimal to condition that $|S \cap A|$ is $\lceil k/2 \rceil - 1$ or $\lceil k/2 \rceil$. Indeed, assume $A \cap B = \emptyset$. Then, regardless of $A \cap S$, the probability is at most $O(1/\sqrt{k})$ that $|S \cap B| \in \{\lceil k/2 \rceil - 1, \lceil k/2 \rceil\}$. Thus, to solve this case, we would have to take other intersection sizes as well, and that would make the analysis much more complicated.

Instead of sticking to this *ad hoc* solution, we use an approach that is guaranteed to be tight. Without loss of generality, we may assume that the optimal solution Γ to the adversary lower bound (2.3) is symmetric with respect to permuting the elements of $[n]$. Then, the matrix Γ can be uniquely described by $k+1$ real numbers. We use representation theory of the symmetric group and obtain necessary and sufficient conditions that these numbers must satisfy. A feasible solution to the dual problem again gives a quantum query algorithm.

Unfortunately, the resulting optimization problem is still very complicated. We were able to obtain a feasible solution, when h is the majority or the exact-half function, using that these functions are symmetric about the weight $k/2$. But applying these scheme for the OR function, for instance, would be much more complicated than our previous *ad hoc* solution. Our solutions for majority and exact-half are essentially equivalent, but for exact-half, the solution turns out to be tight. Generalizing this solution to the exact-ℓ or the ℓ-threshold function is an open problem.

2. Preliminaries

We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$, and 2^A to denote the set of subsets of A. A k-subset is a subset of size k.

All matrices in the paper have real entries. A^* denotes the adjoint (transposed) matrix of A. If A is a matrix, by $A[\![i, j]\!]$, we denote the element on the intersection of row i and column j. By $\|A\|$ we denote the spectral norm of A (the maximal singular value), and by $\|A\|_{\mathrm{tr}}$ we denote the trace norm of A (the sum of the singular values). By $\langle A, B \rangle$ we denote the inner product between the matrices: $\langle A, B \rangle = \mathrm{tr}(A^* B)$.

We assume familiarity with basic probability theory, and we repeatedly use the following well-known result about binomial coefficients:

LEMMA 2.1. *If n and k are positive integers satisfying $k = O(\sqrt{n})$, then $\binom{n}{\lfloor n/2 \rfloor \pm k} = \Theta(2^n/\sqrt{n})$.*

Quantum query complexity Now we define quantum query complexity both in its standard and non-adaptive variants. For a more complete treatment refer to Buhrman & de Wolf (2002) for query complexity and Montanaro (2010) for non-adaptive query complexity. A quantum query algorithm is defined as a sequence of unitary transformations alternated with the oracle calls:

$$U_0 \to O_x \to U_1 \to O_x \to \cdots \to U_{T-1} \to O_x \to U_T. \tag{2.2}$$

Here U_is are arbitrary unitary transformations independent of the input. The oracle O_x is the same in all places, and it depends on the input string $x = (x_i)$ as $|i\rangle_{\mathsf{i}}|b\rangle_{\mathsf{v}} \mapsto |i\rangle_{\mathsf{i}}|b + x_i\rangle_{\mathsf{v}}$ where the addition is performed modulo 2. Other registers besides i and v are left intact. The computation starts in a predefined state $|0\rangle$. After all the operations in (2.2) are performed, some predefined output register is measured. We say that the algorithm evaluates a function f if, for any x in the domain, the result of the measurement is $f(x)$ with probability at least $2/3$. The number T is the *query complexity* of the algorithm. The smallest value of T among all algorithms evaluating f is the quantum query complexity of f, and is denoted by $Q(f)$.

Thus, we see that a quantum algorithm can prepare the input to the next oracle query depending on the results of the previous oracle calls. In many cases, this is crucial for obtaining a good algorithm. But, in some cases, the input to the oracle does not depend on the output of its previous executions. This is captured by the notion of *non-adaptive* quantum query complexity. In such an algorithm, we assume that all the oracle calls happen simultaneously in parallel. More formally, a non-adaptive quantum query algorithm is of the form $U_0 \to O_x^{\otimes T} \to U_1$. The non-adaptive quantum query complexity of f is then defined similarly to the adaptive case.

Formulation of the problem Let us rigorously define our version of the learning problem. Let $h\colon \{0,1\}^k \to \{0,1\}$ be a symmetric Boolean function. It is uniquely defined by a subset $W_h \subseteq \{0,\dots,k\}$ such that $h(x) = 1$ iff $|x| \in W_h$, where $|x|$ stands for the Hamming weight of x. Let $n \geq k$ be a positive integer, and $\mathcal{C}$ denote the set of all k-subsets of $[n]$. If $A \in \mathcal{C}$, we define the function $f_A : \{0,1\}^n \to \{0,1\}$ by $f_A(x) = h(x_A)$ where x_A is the restriction of the input string x to the positions in A. It is more convenient to identify the input string x with the subset $S \subseteq [n]$ defined by $i \in S$ iff $x_i = 1$. Thus, $f_A(S) = 1$ iff $|A \cap S| \in W_h$.

The problem $L_h^n\colon \{0,1\}^{\{0,1\}^n} \to 2^{[n]}$ is defined by $L_h^n(f_A) = A$. Thus, h is fixed and known to the learner in advance, the inputs are the functions f_A (which can be identified with the elements of $\mathcal{C}$), and the input variables are the input strings to f_A (which can be identified with the subsets of $[n]$).

It is easy to see that the quantum query complexity $Q(L_h^n)$ is a non-decreasing function in n. There also exists an upper bound on $Q(L_h^n)$ independent of n. For instance, one may take the complexity of the Fourier sampling algorithm like in Atıcı & Servedio (2007), since its behavior does not depend on n. Hence, there exists $\lim_{n\to\infty} Q(L_h^n)$, which we denote by $Q(L_h)$, and which we are mostly interested in.

Adversary Bound Next, we define the adversary bound tailored to our special case of L_h^n. An adversary matrix Γ is a $\mathcal{C} \times \mathcal{C}$ real symmetric matrix with zeroes along the diagonal. Introducing an abuse of notation, let $\Gamma \circ \Delta_S$ denote the submatrix of Γ formed by the rows in $\{A \in \mathcal{C} \mid f_A(S) = 0\}$ and the columns in $\{B \in \mathcal{C} \mid f_B(S) = 1\}$.

The adversary bound $\mathrm{ADV}^{\pm}(L_h^n)$ is equal to the (common) optimal value of the following two optimization problems:

$$\text{(2.3a)} \qquad \text{maximize} \quad \|\Gamma\|$$

$$\text{(2.3b)} \qquad \text{subject to} \quad \|\Gamma \circ \Delta_S\| \leq 1 \quad \text{for all } S \subseteq [n];$$

$$\text{(2.3c)} \qquad \Gamma[\![A, A]\!] = 0 \quad \text{for all } A \in \mathcal{C}.$$

and

(2.4a)
$$\text{minimize} \quad \max_{A\in\mathcal{C}} \sum\nolimits_{S\subseteq[n]} X_S[\![A,A]\!]$$
(2.4b)
$$\text{subject to} \quad \sum\nolimits_{S\colon f_A(S)\neq f_B(S)} X_S[\![A,B]\!] = 1 \quad \text{for all } A\neq B \text{ in } \mathcal{C};$$
(2.4c)
$$X_S \succeq 0 \qquad \text{for all } S\subseteq[n],$$

where X_S are $\mathcal{C}\times\mathcal{C}$ positive semi-definite matrices (see Reichardt 2009, Theorem 6.2 for the proof of the equality of both problems). The adversary bound is very useful because of the following result:

Theorem 2.5 (Høyer *et al.* 2007; Lee *et al.* 2011). *The quantum query complexity of a function f equals $\Theta(\mathrm{ADV}^{\pm}(f))$.*

Using this theorem, we can estimate $\mathrm{ADV}^{\pm}(L_h)$ instead of $Q(L_h)$. Here we denote $\mathrm{ADV}^{\pm}(L_h) = \lim_{n\to\infty}\mathrm{ADV}^{\pm}(L_h^n)$. The limit exists because $\mathrm{ADV}^{\pm}(L_h^n)$ is a non-decreasing function in n.

An important special case is the positive-weighted adversary, which we denote by $\mathrm{ADV}(L_h^n)$. It is a slight modification of the original version by Ambainis (2002). It is strictly weaker than the general bound, but it is usually much easier to apply. The positive-weighted adversary is defined as in (2.3) and (2.4) with the following modifications. In (2.3), we require all the entries of Γ to be nonnegative. In (2.4), we replace condition (2.4b) by the following one (Špalek & Szegedy 2006, Eq. (3.7)):

(2.6)
$$\sum\nolimits_{S\colon f_A(S)\neq f_B(S)} X_S[\![A,B]\!] \geq 1 \qquad \text{for all } A\neq B \text{ in } \mathcal{C};$$

3. Combinatorial group testing

In this section, we describe a quantum query algorithm for the combinatorial group testing problem. We solve the problem in its original form, which deviates slightly from our version of the learning problem. Let us reformulate the problem. Let $k<n$ be fixed positive integers, and $\mathcal{C}$ consist of all subsets of $[n]$ of sizes at

most k. For each $A \in \mathcal{C}$, the function $f_A \colon 2^{[n]} \to \{0,1\}$ is defined by

$$f_A(S) = \begin{cases} 1, & \text{if } A \cap S \neq \emptyset; \\ 0, & \text{otherwise.} \end{cases}$$

We are given oracle access to f_A, and the task is to detect A. The difference with the L_{OR} problem is that we allow A of size less than k. In this section, we prove the following result:

THEOREM 3.1. *The quantum query complexity of the combinatorial group testing problem is* $\Theta(\sqrt{k})$.

The lower bound can be proved by a reduction from the unordered search, refer to Ambainis & Montanaro (2014) for more detail. Here we prove the upper bound. We do so by constructing a feasible solution to (2.4). This is done in two steps: First, we define rank-1 matrices $Y_S(p)$, and then build the matrices X_S from them.

Let P be the binomial probability distribution on $[n]$ with probability p. Recall that it is a probability distribution on the subsets of $[n]$, where each element of $[n]$ is included into the subset independently with probability p. By $P(S)$, we denote the probability of sampling S from P: $P(S) = p^{|S|}(1-p)^{n-|S|}$. Finally, let $\triangle$ denote the symmetric difference of sets.

We define $Y(p) = (Y_S(p))_{S \subseteq [n]}$ by

$$Y_S(p) = \frac{P(S)}{2p}\, \psi\psi^* \succeq 0,$$

where

$$\psi[\![A]\!] = \frac{1}{(1-p)^{|A|/2}} \times \begin{cases} \sqrt[4]{kp/(1-p)}, & \text{if } |A \cap S| = 0; \\ \sqrt[4]{(1-p)/(kp)}, & \text{if } |A \cap S| = 1; \\ 0, & \text{otherwise;} \end{cases}$$

for all $A \in \mathcal{C}$. In this notation,

$$\sum_{S\subseteq[n]} Y_S(p)[\![A,A]\!] = \frac{1}{2p\,(1-p)^{|A|}}$$
$$\times\left(\Pr_{S\sim P}\big[|S\cap A|=0\big]\sqrt{\frac{kp}{1-p}} \;+\; \Pr_{S\sim P}\big[|S\cap A|=1\big]\sqrt{\frac{1-p}{kp}}\,\right)$$
$$= \frac{1}{2p\,(1-p)^{|A|}}\left((1-p)^{|A|}\sqrt{\frac{kp}{1-p}} \;+\; |A|p\,(1-p)^{|A|-1}\sqrt{\frac{1-p}{kp}}\,\right)$$
$$\leq \sqrt{\frac{k}{p\,(1-p)}}\,.$$

Now we fix two distinct elements A, B of $\mathcal{C}$. An element A is used in Y_S only if $|S\cap A| \leq 1$. Thus, we are only interested in $S\subseteq[n]$ such that $|A\cap S| + |B\cap S| = 1$. Thus,

$$\sum_{S\colon f_A(S)\neq f_B(S)} Y_S(p)[\![A,B]\!] = \frac{\Pr_{S\sim P}\big[|A\cap S| + |B\cap S| = 1\big]}{2p\,(1-p)^{(|A|+|B|)/2}}$$
$$= \frac{|A\triangle B|\,p\,(1-p)^{|A\cup B|-1}}{2p\,(1-p)^{(|A|+|B|)/2}} = \frac{|A\triangle B|}{2}(1-p)^{\frac{|A\triangle B|}{2}-1}\,.$$

Now, for each $S\subseteq[n]$, let

$$X_S = \int_0^1 Y_S(p)\,\mathrm{d}p\,.$$

First, each X_S is positive semi-definite, because positive semi-definite matrices form a convex cone. Next, for any $A\in\mathcal{C}$:

$$\sum_{S\subseteq[n]} X_S[\![A,A]\!] \leq \sqrt{k}\int_0^1 \frac{\mathrm{d}p}{\sqrt{p\,(1-p)}} = \pi\sqrt{k}\,.$$

And finally, for all $A\neq B$ in $\mathcal{C}$:

$$\sum_{S\colon f_A(S)\neq f_B(S)} X_S[\![A,B]\!] = \frac{|A\triangle B|}{2}\int_0^1 (1-p)^{\frac{|A\triangle B|}{2}-1}\,\mathrm{d}p = 1.$$

4. Application of representation theory

In the previous section, we described an ad hoc construction of a feasible solution to (2.4) when h is the OR function. In this section, we use representation theory to give an alternative description for $\mathrm{ADV}^{\pm}(L_h)$ that works for any function h. We work with the lower bound (2.3), because it has a very simple structure. In the next two sections, we use duality to the new formulation to prove that the quantum query complexity of the $L_{\text{EXACT-HALF}_k}$ and the L_{MAJORITY_k} problems is $O(k^{1/4})$.

Let $h : \{0,1\}^k \to \{0,1\}$ be a symmetric function defined by the subset W_h of weights, i.e., $h(x) = 1$ iff $|x| \in W_h$. The search for an adversary matrix for the function L_h turns out to be equivalent to the search for a list of real numbers $d = (d_0, \ldots, d_k)$ satisfying the constraints we are about to describe.

Let $m \le k$ be a positive integer and $0 < p < 1$ be a real number. We make use of Krawtchouk polynomials for probability p. These polynomials are orthogonal with respect to the binomial distribution (see Szegő 1975 for the general definition, and Krasikov & Litsyn 2001 for the special case $p = 1/2$, which we use in Sections 5 and 6). We treat them as column vectors in $\mathbb{R}^{m+1}$ and also include the weight (due to the weight, they cease to be polynomials). With this modification, the definition is as follows:

(4.1)

$$K_t^{(m,p)}[\![x]\!] = \sqrt{\binom{m}{x} p^x (1-p)^{m-x}} \sum_{i=0}^{t} (-1)^i p^{t-i} (1-p)^i \binom{x}{i} \binom{m-x}{t-i},$$

where $t, x \in \{0, \ldots, m\}$. Let $\varkappa_t^{(m,p)} = K_t^{(m,p)} / \|K_t^{(m,p)}\|$ be the corresponding normalized vectors. Thus, $\{\varkappa_t^{(m,p)}\}$, for fixed m and p, form an orthonormal basis of $\mathbb{R}^{m+1}$. We use the list d to define the matrices

$$M_{m,p}^{(d)} = \sum_{i=0}^{m} d_{k-i} \varkappa_{m-i}^{(m,p)} \big(\varkappa_{m-i}^{(m,p)}\big)^*. \tag{4.2}$$

Let $0 \le t \le k - m$ be an integer, and define $W_1(t) = \{\ell \in \mathbb{Z} \mid 0 \le \ell \le m,\ \ell + t \in W_h\}$, and $W_0(t) = \{0, \ldots, m\} \setminus W_1(t)$. Let

$$M^{(d)}_{m,p,t} = M^{(d)}_{m,p}[\![W_0(t), W_1(t)]\!] \tag{4.3}$$

be the submatrix of $M^{(d)}_{m,p}$ formed by the rows in $W_0(t)$ and the columns in $W_1(t)$.

The aim of this section is to prove the following result:

THEOREM 4.4. *For any symmetric function h, $\mathrm{ADV}^{\pm}(L_h)$ equals the supremum of $\max_i d_i$ over all lists of reals $d = (d_0, \ldots, d_k)$ satisfying the following constraints:*

- *$d_k = 0$, and*
- *for all integers $0 < m \le k$, $0 \le t \le k-m$, and reals $0 < p < 1$, we have $\|M^{(d)}_{m,p,t}\| \le 1$, where $M^{(d)}_{m,p,t}$ is defined in (4.3).*

In order to prove this theorem, we need some basic results from representation theory of the symmetric group. These results are only used in this section. The reader may refer to a textbook on the topic like, e.g., Sagan (2001), or to the appendix, where we briefly formulate the required notions and results.

If N is a finite set, let us denote by $\mathbb{S}_N$ the symmetric group on N. We consider modules over the group algebra $\mathbb{R}G$ where G is either a symmetric group or a direct product of two symmetric groups.

Fix an integer n, and consider the problem L^n_h. Let also $N = [n]$. The rows and the columns of an adversary matrix Γ are labeled by k-subsets of N. The problem is symmetric with respect to the permutations of variables, so by Høyer *et al.* (2007) we may assume that Γ is symmetric with respect to $\mathbb{S}_N$. More specifically, Γ does not change if we simultaneously transform the labels of its rows and columns by $\{a_1, \ldots, a_k\} \mapsto \{\pi a_1, \ldots, \pi a_k\}$ for some $\pi \in \mathbb{S}_N$.

The real vector space with the set of k-subsets of N as its orthonormal basis, and the above action of $\mathbb{S}_N$, is the permutation $\mathbb{R}\mathbb{S}_N$-module corresponding to the partition $(n-k, k)$ of n. We denote it by $M(N, k)$. We denote the basis element of $M(N, k)$ corresponding to A by A itself.

Now consider $\|\Gamma \circ \Delta_S\|$ for $S \subseteq N$. We denote $N_0 = N \setminus S$, $N_1 = S$, $n_0 = |N_0|$, and $n_1 = |N_1|$. Then, $\Gamma \circ \Delta_S$ is symmetric with respect to $\mathbb{S}_{N_0} \times \mathbb{S}_{N_1}$. Thus, we have to understand how the $\mathbb{RS}_N$-module $M(N,k)$ behaves under restriction to this subgroup. It is easy to see that

$$M(N,k)\downarrow_{\mathbb{S}_{N_0}\times\mathbb{S}_{N_1}} = \bigoplus_{k_0+k_1=k} M(N_0,k_0) \otimes M(N_1,k_1), \tag{4.5}$$

where $A \otimes B$, with A being a basis element of $M(N_0,k_0)$ and B being a basis element of $M(N_1,k_1)$, is understood as the basis element $A \cup B$ of $M(N,k)$. We continue using the convention that $A \otimes B$ is the disjoint union of A and B later, for instance, in (4.8).

Let Π_1 be the projector onto the spaces on the right-hand side of (4.5) with $k_1 \in W_h$, and Π_0 be the projector onto the orthogonal complement of this space. Then,

$$\|\Gamma \circ \Delta_S\| = \|\Pi_0 \Gamma \Pi_1\|. \tag{4.6}$$

The following result describes the decomposition of $M(N,k)$ into irreducible submodules. They are isomorphic to the Specht modules $S(N,t)$ corresponding to partitions $(n-t,t)$ of n. The modules with different values of t are not isomorphic. The lemma follows from general theory (Sagan 2001, Sections 2.9 and 2.10). We give a proof in the appendix.

Lemma 4.7. *The $\mathbb{RS}_N$-module $M(N,k)$ has the following decomposition into irreducible submodules: $M(N,k) = \bigoplus_{t=0}^{k} S_k(N,t)$, where each $S_k(N,t)$ is isomorphic to $S(N,t)$. The submodule $S_k(N,t)$ is spanned by the vectors*

$$\begin{aligned} v_k(N,t,a,b) = {} & \big(\{a_1\} - \{b_1\}\big) \otimes \cdots \otimes \big(\{a_t\} - \{b_t\}\big) \\ & \otimes \Big(\sum_{A \subseteq N \setminus \{a_1,\ldots,a_t,b_1,\ldots,b_t\} \colon |A| = k-t} A \Big) \end{aligned} \tag{4.8}$$

defined by disjoint sequences $a = (a_1,\ldots,a_t)$ and $b = (b_1,\ldots,b_t)$ of pairwise distinct elements of N. The dimension of $S(N,t)$ is $\binom{n}{t} - \binom{n}{t-1}$. Moreover, the only (up to a scalar) $\mathbb{RS}_N$-isomorphism of $S_k(N,t)$ onto $S_\ell(N,t)$ maps the vector $v_k(N,t,a,b)$ into a scalar multiple of $v_\ell(N,t,a,b)$ for any choice of a and b.

We define $S_{k_0}(N_0, t_0)$ and $S_{k_1}(N_1, t_1)$ similarly. By combining (4.5) and Lemma 4.7, we get that the irreducible $\mathbb{R}(\mathbb{S}_{N_0} \times \mathbb{S}_{N_1})$-submodules of $M(N,k)\downarrow_{\mathbb{S}_{N_0}\times\mathbb{S}_{N_1}}$ are $S_{k_0}(N_0,t_0) \otimes S_{k_1}(N_1,t_1)$, where

$$k_0 + k_1 = k, \quad 0 \le t_0 \le k_0, \quad \text{and} \quad 0 \le t_1 \le k_1. \tag{4.9}$$

Two submodules of this form are isomorphic iff their values of t_0 and t_1 are equal. Thus,

$$R(t_0, t_1) = \bigoplus_{k_0,\, k_1 \text{ satisfy (4.9)}} S_{k_0}(N_0, t_0) \otimes S_{k_1}(N_1, t_1),$$

are the canonical submodules of $M(N,k)\downarrow_{\mathbb{S}_{N_0}\times\mathbb{S}_{N_1}}$ with multiplicities $k + 1 - t_0 - t_1$.

By Schur's lemma, in a suitable basis of $R(t_0, t_1)$, any $\mathbb{R}(\mathbb{S}_{N_0} \times \mathbb{S}_{N_1})$-homomorphism from $R(t_0, t_1)$ to itself is of the form $A \otimes I_{t_0,t_1}$, where A is an $(k+1-t_0-t_1)\times(k+1-t_0-t_1)$ matrix, and I_{t_0,t_1} is the identity matrix in $S(N_0,t_0)\otimes S(N_1,t_1)$. For each (t_0, t_1), we choose the basis $\{e_\ell\}_{\ell\in\{0,\ldots,k-t_0-t_1\}}$ for the matrix A so that $(e_\ell e_\ell^*) \otimes I_{t_0,t_1}$ projects onto $S_{k-t_1-\ell}(N_0, t_0) \otimes S_{t_1+\ell}(N_1, t_1)$. With this choice of the basis, we have that

$$\Pi_0(A \otimes I_{t_0,t_1})\Pi_1 = A[\![W_0(t_1), W_1(t_1)]\!] \otimes I_{t_0,t_1}, \tag{4.10}$$

where Π_0 and Π_1 are as in (4.6), and W_0 and W_1 are as in (4.3).

Let $\Pi_k(N,t)$ denote the orthogonal projector onto $S_k(N,t)$. Again, by Schur's lemma,

$$\Pi_k(N,t) = \bigoplus_{t_0,t_1} A^{(t)}_{t_0,t_1} \otimes I_{t_0,t_1}$$

for some matrices $A^{(t)}_{t_0,t_1}$. By the Littlewood-Richardson rule (Sagan 2001, Section 4.9),

$$S(N,t)\downarrow_{\mathbb{S}_{N_0}\times\mathbb{S}_{N_1}} \cong \bigoplus_{t_0+t_1\le t} S(N_0,t_0) \otimes S(N_1,t_1), \tag{4.11}$$

so $A^{(t)}_{t_0,t_1}$ is zero if $t < t_0+t_1$, and, otherwise, it is a rank-1 orthogonal projector (as the corresponding multiplicity is 1).

At the heart of the proof of Theorem 4.4 is the following observation (recall that the matrices $A^{(t)}_{t_0,t_1}$ depend on the values of n and n_1):

LEMMA 4.12. *For any $0 < p < 1$, the projector $A^{(t)}_{t_0,t_1}$ tends to the projector onto $\varkappa^{(k-t_0-t_1,p)}_{t-t_0-t_1}$ as $n \to \infty$ and $n_1/n \to p$. Moreover, the convergence is uniform for $c < p < 1-c$ where $c > 0$ is any constant. On the other hand, there exists a bound ε_c satisfying $\lim_{c\to 0} \varepsilon_c = 0$, such that $\|A^{(t)}_{t_0,t_1} - e_{t-t_0-t_1}e^*_{t-t_0-t_1}\| \le \varepsilon_c$ if n_1 is less than cn, and $\|A^{(t)}_{t_0,t_1} - e_{k-t}e^*_{k-t}\| \le \varepsilon_c$ if n_1 is more than $(1-c)n$.*

We prove the lemma at the end of the section. For now, let us show how the lemma can be used to prove Theorem 4.4.

Assume that $\mathrm{ADV}^{\pm}(L_h) = Q$. As noticed in Section 2, $Q < \infty$. Then, for each n, let $\Gamma^{(n)}$ be an optimal solution to (2.3). We may assume that $\|\Gamma^{(n)}\|$ is an eigenvalue of $\Gamma^{(n)}$, otherwise replacing $\Gamma^{(n)}$ by $-\Gamma^{(n)}$. By Schur's lemma, we may also assume that

$$\Gamma^{(n)} = \sum_{t=0}^{k} d^{(n)}_t \Pi_k(N,t). \tag{4.13}$$

Consider the vectors $d^{(n)} = (d^{(n)}_t)$. As the absolute values of all $d^{(n)}_t$ are bounded by Q, the Bolzano-Weierstrass theorem gives a convergent subsequence $d^{(n_1)}, d^{(n_2)}, \ldots$. We define $d = (d_t)$ as the limit of this subsequence. Clearly, $\max_t d_t = Q$.

Next, $\operatorname{tr}\Pi_k(N,t) = \binom{n}{t} - \binom{n}{t-1}$. Hence, $\operatorname{tr}\Pi_k(N,k)$ overwhelms the traces of all other projectors in (4.13) as $n \to \infty$. Thus, by (2.3c),

$$d_k = \lim_{i\to\infty} d^{(n_i)}_k = \lim_{i\to\infty} \frac{\operatorname{tr}\Gamma^{(n_i)}}{\binom{n_i}{k} - \binom{n_i}{k-1}} = 0. \tag{4.14}$$

This proves the first constraint in Theorem 4.4. The second constraint follows from Lemma 4.12 and (4.10).

Now assume d is an optimal solution to the optimization problem in Theorem 4.4, and let $Q = \max_t d_t$. We define $\Gamma^{(n)}$ as in (4.13), where $d^{(n)}_t = d_t$ for $t < k$, and $d^{(n)}_k$ is chosen so that $\operatorname{tr}(\Gamma^{(n)}) = 0$. Then, due to symmetry, all diagonal entries of $\Gamma^{(n)}$ are equal to zero. Also, similarly to (4.14), $\lim_{n\to\infty} d^{(n)}_k = 0$.

Choose $c > 0$ so that $\varepsilon_c \le 1/(2(k+1)Q)$, where ε_c is as in Lemma 4.12. If $|S|/n < c$ or $|S|/n > 1-c$, then $\|\Gamma^{(n)} \circ \Delta_S\| \le 1/2$ for *any* choice of d satisfying $\max_t d_t \le Q$. If $|S|/n \to p$ with

$c < p < 1 - c$, then $\lim_{n\to\infty} \|\Gamma^{(n)} \circ \Delta_S\| = 1$ by Lemma 4.12 and (4.10) again.

Proof of Lemma 4.12. Fix two sequences of pairwise distinct elements in N_0: $a = (a_1, \ldots, a_{t_0})$ and $b = (b_1, \ldots, b_{t_0})$, and two sequences $a' = (a'_1, \ldots, a'_{t_1})$ and $b' = (b'_1, \ldots, b'_{t_1})$ in N_1. In order to find the vector onto which $A^{(t)}_{t_0,t_1}$ projects, it suffices to find a linear combination of the vectors

$$\left\{ v_{k_0}(N_0, t_0, a, b) \otimes v_{k_1}(N_1, t_1, a', b') \;\middle|\; k_0,\, k_1 \text{ satisfy (4.9)} \right\}$$

that belongs to $S_k(N, t)$.

Clearly, $(\{a_1\} - \{b_1\}) \otimes \cdots \otimes (\{a_{t_0}\} - \{b_{t_0}\})$ and $(\{a'_1\} - \{b'_1\}) \otimes \cdots \otimes (\{a'_{t_1}\} - \{b'_{t_1}\})$ factor out in any linear combination, so we can remove the elements in a, b, a' and b', and consider the case $t_0 = t_1 = 0$. The removal has the effect that t gets reduced by $t_0 + t_1$, k_0 by t_0, k_1 by t_1, n_0 by $2t_0$, and n_1 by $2t_1$. The effect on t, k_0 and k_1 is reflected in the statement of the lemma, and the change in n_0 and n_1 is not substantial, as we assume $n \to \infty$.

So, it suffices to consider the case $t_0 = t_1 = 0$. In this case, the vector

$$\frac{1}{t!} \sum_{a,b} v_k(N, t, a, b), \tag{4.15}$$

where the sum is over all sequences a in N_0 and b in N_1, is a linear combination of the vectors

$$\left\{ v_{k_0}(N_0, 0, \emptyset, \emptyset) \otimes v_{k_1}(N_1, 0, \emptyset, \emptyset) \;\middle|\; k_0 + k_1 = k \right\}.$$

More specifically, the coefficient of $v_{k_0}(N_0, 0, \emptyset, \emptyset) \otimes v_{k_1}(N_1, 0, \emptyset, \emptyset)$ in (4.15) is

$$\sum_{i=0}^{t} (-1)^i \binom{k_0}{t-i} \binom{k_1}{i} (n_0 - k_0)^{\underline{i}}\, (n_1 - k_1)^{\underline{t-i}}\,, \tag{4.16}$$

where $a^{\underline{b}} = a(a-1)\cdots(a-b+1)$ denotes the falling power. That is, we claim that if A is a k_0-subset of N_0 and B is a k_1-subset of N_1, then the coefficient of $A \otimes B$ in (4.15) is (4.16). Indeed, i is

the number of the elements of B used in the sequence b, $\binom{k_1}{i}$ is the number of ways to choose them, $(n_0-k_0)^{\underline{i}}$ is the number of ways to choose the elements of $N_0 \backslash A$ that serve as the corresponding element of the sequence a (they do not appear in the product). Calculations for A are similar.

Taking the norm of the vector $v_{k_0}(N_0,0,\emptyset,\emptyset) \otimes v_{k_1}(N_1,0,\emptyset,\emptyset)$ into account, we get that $A^{(t)}_{0,0}$ projects onto the vector $w_t \in \mathbb{R}^{k+1}$ defined by (where we assumed $\ell = k_1$):

$$w_t[\![\ell]\!] = \sqrt{\binom{n_1}{\ell}\binom{n_0}{k-\ell}} \times \sum_{i=0}^{t} (-1)^i \binom{\ell}{i}\binom{k-\ell}{t-i}(n_1-\ell)^{\underline{t-i}}\,(n_0-k+\ell)^{\underline{i}}.$$

Let $\widetilde{w}_t = w_t/\|w_t\|$. Assuming $n \to \infty$ and $n_1/n \to p$, it is easy to check that $\widetilde{w}_t \to \varkappa_t^{(k,p)}$. Also, the convergence is uniform if $c < p < 1-c$.

Notice that the largest power of n_0 in $w_t[\![\ell]\!]$ is $(k-\ell)/2 + \min\{\ell,t\}$, and the largest power of n_1 is $\ell/2+\min\{k-\ell,t\}$. Hence, $\widetilde{w}_t$ is close to e_t if n_1/n is sufficiently small. Also, $\widetilde{w}_t$ is close to e_{k-t} if n_0/n is small. □

5. Exact-half function

In this section, we apply Theorem 4.4 to the exact-half function. The function EXACT-HALF$_k\colon \{0,1\}^k \to \{0,1\}$ is equal to 1 iff the input string has Hamming weight $\lfloor k/2 \rfloor$ exactly. This section is devoted to the proof of the following result:

Theorem 5.1. *The quantum query complexity of $L_{\text{EXACT-HALF}_k}$ is $\Theta(k^{1/4})$.*

The lower bound can be shown using a simple positive-weighted adversary. Consider the adversary matrix Γ for $L^n_{\text{EXACT-HALF}_k}$ defined by $\Gamma[\![A,B]\!] = 1$ if $A \neq B$ and $\Gamma[\![A,A]\!] = 0$. We have $\|\Gamma\| = \binom{n}{k} - 1$. On the other hand, $\Gamma \circ \Delta_S$ is the all-1 matrix, and a simple argument involving Lemma 2.1 shows that it has $O\big(\binom{n}{k}/\sqrt{k}\big)$ columns. Also, $\Gamma \circ \Delta_S$ still has almost $\binom{n}{k}$ rows, hence,

$\|\Gamma \circ \Delta_S\| = O\big(\binom{n}{k} k^{-1/4}\big)$. Thus, $\mathrm{ADV}(L_h^n) = \Omega(k^{1/4})$. In the remaining part of this section, we show that this simple lower bound is actually tight.

We do so by providing a feasible solution to the optimization problem in Theorem 4.4. Note that this optimization problem has the following self-reducibility property: For every $k' < k$, if we denote $d'_i = d_{i+k-k'}$ and take an appropriate subset of the constraints, we obtain the optimization problem for the case $h = \text{EXACT-HALF}_{k'}$. This has a number of consequences. The first one is that it suffices to estimate d_0 only, because d_i with larger values of i have been already estimated for smaller values of k.

We consider the constraints $\left\| M^{(d)}_{m,1/2,\lfloor k/2\rfloor-\lfloor m/2\rfloor} \right\| \leq 1$, where m ranges from 1 to k. Let,

$$A_{m,\ell} = \big(\varkappa_\ell^{(m,1/2)} (\varkappa_\ell^{(m,1/2)})^*\big) [\![W_0(t), W_1(t)]\!] \tag{5.2}$$

in the notation of (4.2) and (4.3), where $t = \lfloor k/2 \rfloor - \lfloor m/2 \rfloor$. With this choice of parameters, $A_{m,\ell}$ is an $m \times 1$ matrix. Later in the proof, we will treat it as a vector in $\mathbb{R}^m$. Note also that the matrices $A_{m,\ell}$ do *not* depend on k. Thus, we get the following optimization problem:

(5.3a)

$$\text{maximize} \quad d_0$$

(5.3b)

$$\text{subject to} \quad \left\| \sum_{i=1}^{m} d_{k-i} A_{m,m-i} \right\| \leq 1 \qquad \text{for all } m = 1, \ldots, k;$$

(5.3c)

$$d_i \in \mathbb{R} \qquad \text{for } i = 0, \ldots, k-1.$$

Applying semi-definite duality (Boyd & Vandenberghe 2004, Section 5.9), we obtain the following upper bound on (5.3):

$$\text{minimize} \quad \sum_{m=1}^{k} \|\Lambda_m\|_{\mathrm{tr}} \tag{5.4a}$$

$$\text{subject to} \quad \langle \Lambda_k, A_{k,0} \rangle = 1; \tag{5.4b}$$

$$\sum_{i=0}^{\ell} \langle \Lambda_{k-i}, A_{k-i,\ell-i} \rangle = 0 \qquad \text{for all } \ell; \tag{5.4c}$$

$$\Lambda_m \text{ has the same size as } A_{m,\ell} \qquad \text{for all } m \text{ and } \ell. \tag{5.4d}$$

We are going to construct a feasible solution to this problem. We use the following elimination strategy. Constraint (5.4b) only uses Λ_k. So, we take some Λ_k that satisfies $\langle \Lambda_k, A_{k,0} \rangle = 1$, but may have nonzero inner products with other $A_{k,i}$. Then we take Λ_{k-1} that satisfies (5.4c) for $\ell = 1$, then Λ_{k-2} that satisfies (5.4c) for $\ell = 2$, and so on. We find Λ_{k-i} for $i > 0$ using self-reducibility.

More formally, we apply induction. Let $\Lambda^{(k)} = (\Lambda_1^{(k)}, \ldots, \Lambda_k^{(k)})$ be our solution to (5.4) for a specific value of k, and let $g(k)$ denote the corresponding value of (5.4a). The base case, $\Lambda^{(1)}$, is trivial to construct. Assume we have constructed $\Lambda^{(k')}$ for all $k' < k$. Then, we take

$$\Lambda^{(k)} = (0, \ldots, 0, \Lambda_k) - \sum_{\ell=1}^{k-1} \langle \Lambda_k, A_{k,\ell} \rangle \, \Lambda^{(k-\ell)},$$

where the first list has Λ_k in the kth position, and the remaining lists are padded with zeroes from the right. Here Λ_k is some matrix satisfying (5.4b). We will define it later. It is easy to check that $\Lambda^{(k)}$ satisfies (5.4b) and (5.4c). Using the triangle inequality for the trace norm, we obtain

$$g(k) \leq \|\Lambda_k\|_{\mathrm{tr}} + \sum_{\ell=1}^{k-1} \left| \langle \Lambda_k, A_{k,\ell} \rangle \right| \, g(k-\ell). \tag{5.5}$$

So, it remains to choose Λ_k. For the remainder of this section and the next section, let $\varkappa_\ell = \varkappa_\ell^{(k,1/2)}$. Recall that $\{\varkappa_\ell\}$ form an orthonormal basis of $\mathbb{R}^{k+1}$. Let, for brevity, $s = \lfloor k/2 \rfloor$. We have $A_{k,\ell} = \varkappa_\ell[\![s]\!] \, \breve{\varkappa}_\ell$, where $\breve{\varkappa}_\ell$ denotes $\varkappa_\ell$ with the sth element removed. We take

$$\Lambda_k = \frac{1}{\varkappa_0[\![s]\!](1 - \varkappa_0[\![s]\!]^2)} \, \breve{\varkappa}_0.$$

It is straightforward to check that $\langle \Lambda_k, A_{k,0} \rangle = 1$. Also, for $\ell > 0$, $\langle \breve{\varkappa}_0, A_{k,\ell} \rangle = \varkappa_\ell[\![s]\!] \langle \breve{\varkappa}_0, \breve{\varkappa}_\ell \rangle = -\varkappa_0[\![s]\!]\varkappa_\ell[\![s]\!]^2$. Hence,

$$\langle \Lambda_k, A_{k,\ell} \rangle = \frac{-\varkappa_\ell[\![s]\!]^2}{1 - \varkappa_0[\![s]\!]^2}. \tag{5.6}$$

Now we apply additional properties of $\varkappa_\ell$. First, $\varkappa_0[\![x]\!] = \sqrt{\binom{k}{x}/2^k}$. Thus, by Lemma 2.1, $\varkappa_0[\![s]\!] = \Theta(k^{-1/4})$, and $\|\Lambda_k\|_{\mathrm{tr}} = \Theta(k^{1/4})$.

Another property (Krasikov & Litsyn 2001, Eq. (32)) is $\varkappa_\ell[\![s]\!] = \pm\varkappa_{k-\ell}[\![s]\!]$. As $\{\varkappa_\ell\}$ form an orthonormal basis, $\sum_{\ell=0}^{k} \varkappa_\ell[\![s]\!]^2 = 1$, hence, by (5.6),

$$\sum_{\ell=1}^{k-1} |\langle \Lambda_k, A_{k,\ell} \rangle| = \frac{1 - 2\varkappa_0[\![s]\!]^2}{1 - \varkappa_0[\![s]\!]^2},$$

and

$$\frac{1 - \varkappa_0[\![s]\!]^2}{1 - 2\varkappa_0[\![s]\!]^2} \sum_{\ell=1}^{k-1} (k - \ell) |\langle \Lambda_k, A_{k,\ell} \rangle| = \frac{k}{2}.$$

Let C_0 be some constant such that $g(k) \le C_0 k^{1/4}$ for small values of k, and let C_1 be such that $\|\Lambda_k\|_{\mathrm{tr}} \le C_1 k^{1/4}$ for all k. Then, we prove by induction that $g(k) \le C k^{1/4}$ for $C = \max\{C_0, \frac{\sqrt[4]{2}}{\sqrt[4]{2}-1} C_1\}$. Indeed, this is satisfied for the small values of k. Assume this is satisfied for all $k' < k$. Then, by (5.5) and the concavity of $k^{1/4}$:

$$\begin{aligned} g(k) &\le C_1 k^{1/4} + \sum_{\ell=1}^{k-1} |\langle \Lambda_k, A_{k,\ell} \rangle| C(k-\ell)^{1/4} \\ &\le C_1 k^{1/4} + C\left(\frac{1 - \varkappa_0[\![s]\!]^2}{1 - 2\varkappa_0[\![s]\!]^2} \sum_{\ell=1}^{k-1} |\langle \Lambda_k, A_{k,\ell} \rangle| (k-\ell) \right)^{1/4} \\ &= C_1 k^{1/4} + C(k/2)^{1/4} \le C k^{1/4}. \end{aligned}$$

6. Majority function

In this section, we prove some partial results on the quantum query complexity of the L_{MAJORITY_k} function. The function is defined by $\mathrm{MAJORITY}_k(x) = 1$ iff $|x| \ge k/2$. First, the algorithm from Section 5 carries over to this case with minor modifications.

THEOREM 6.1. *The quantum query complexity of* $L_{MAJORITY_k}$ *is* $O(k^{1/4})$.

PROOF. Again, we construct a feasible solution to (5.4) where $A_{m,\ell}$ are as in (5.2) with W_0 and W_1 modified accordingly. This time, we use different strategies to construct $\Lambda^{(k)}$ for odd and even values of k. We need the following easy symmetry result about Krawtchouk polynomials (Krasikov & Litsyn 2001, Eq. (31)):

$$\varkappa_\ell[\![x]\!] = (-1)^\ell \varkappa_\ell[\![k-x]\!], \tag{6.2}$$

where again $\varkappa_\ell = \varkappa_\ell^{(k,1/2)}$. We also use notations $W_0 = W_0(0)$ and $W_1 = W_1(0)$.

For the even values of k, we use the same elimination strategy, but we change the way we define the matrix Λ_k. Let $s = k/2$, and let this time $\breve{\varkappa}_\ell$ denote the $W_0 \times W_1$ matrix having the elements of $\varkappa[\![W_0]\!]$ in column s, and zeroes everywhere else. Intuitively, the nonzero elements of $\breve{\varkappa}_\ell$ form the upper half of the vector $\breve{\varkappa}_\ell$ from the proof of Theorem 5.1. We define

$$\Lambda_k = \frac{2}{\varkappa_0[\![s]\!](1-\varkappa_0[\![s]\!]^2)}\,\breve{\varkappa}_0.$$

From (6.2), we get $\langle \Lambda_k, A_{k,0}\rangle = 1$. Also, $\|\Lambda_k\|_{\mathrm{tr}} = \Theta(k^{1/4})$. If ℓ is odd, we get from (6.2) that the sth column of $A_{k,\ell}$ consists of zeroes. If ℓ is even, using the same property, we get that $\langle \breve{\varkappa}_0, \breve{\varkappa}_\ell\rangle = -\varkappa_0[\![s]\!]\varkappa_\ell[\![s]\!]/2$. Either way, (5.6) holds. The proof further proceeds as in Section 5. Also, we have to note that $\langle \Lambda_k, A_{k,\ell}\rangle = 0$ if ℓ is odd, hence, we only need $\Lambda^{(k')}$ with even values of $k' < k$ to define $\Lambda^{(k)}$.

Now assume that k is odd. We know that $d_1 = O(k^{1/4})$ by considering $\Lambda^{(k-1)}$. Thus, we change our strategy and prove that $d_0 - d_1 = O(1)$. If we replace (5.3a) by $d_0 - d_1$, we get the problem (5.4) with (5.4b) replaced by

$$\langle \Lambda_k, A_{k,0}\rangle = 1 \qquad \text{and} \qquad \langle \Lambda_k, A_{k,1}\rangle + \langle \Lambda_{k-1}, A_{k-1,0}\rangle = -1, \tag{6.3}$$

and ℓ ranging in (5.4c) from 2 to $k-1$. A possible feasible solution is

$$\Lambda_k = \frac{2}{\langle \varkappa_0[\![W_1]\!], \varkappa_1[\![W_1]\!]\rangle}(\varkappa_0[\![W_0]\!])(\varkappa_1[\![W_1]\!])^*$$

and $\Lambda_m = 0$ for other values of m. Using (6.2) and the orthogonality of $\{\varkappa_\ell\}$, we get that $\langle \varkappa_0[\![W_0]\!], \varkappa_\ell[W_0]\rangle = 0$ for even $\ell \geq 2$, and $\langle \varkappa_1[\![W_1]\!], \varkappa_\ell[W_1]\rangle = 0$ for odd $\ell \geq 3$, hence (5.4c) holds. We get (6.3) similarly. Finally, $K_1[\![x]\!] = k - 2x$, $\|K_0\| = 1$ and $\|K_1\| = \sqrt{k}$, where K is defined in (4.1) (Krasikov & Litsyn 2001, Eqs. (12, 33)), thus, using the definition of $\varkappa$ and the central limit theorem:

$$\begin{aligned}\langle \varkappa_0[\![W_1]\!], \varkappa_1[\![W_1]\!]\rangle &= \frac{1}{2^k\sqrt{k}} \sum_{x=\lceil k/2\rceil}^{k} \binom{k}{x}(k-2x) \\ &\xrightarrow{k\to\infty} -\frac{4}{k\sqrt{2\pi}} \int_0^\infty \mathsf{e}^{-2x^2/k} x \,\mathrm{d}x = -\frac{1}{\sqrt{2\pi}}.\end{aligned}$$

Hence, $\|\Lambda_k\|_{\mathrm{tr}} = O(1)$. □

For the case when h is the OR or the exact-half function, we were able to prove tight lower bounds using the positive-weighted adversary. In the next theorem, we show that it is not possible to prove a polynomial (in k) lower bound using this technique, when h is the majority function. There are some limitations known on the positive-weighted adversary, like the certificate complexity barrier (Špalek & Szegedy 2006) or the property testing barrier (Høyer *et al.* 2007). Neither apply here, so we give a direct proof using the optimization problem given by (2.4a), (2.6) and (2.4c).

THEOREM 6.4. *No positive-weighted adversary for the function $L_{MAJORITY_k}$ can be better than $\Omega(\log k)$.*

PROOF. Fix n. If $X = (X_S)$ is a family of positive semi-definite matrices, let $m(X)$ stand for the objective (2.4a), and $\ell_{A,B}(X)$ stand for the left-hand side of (2.6). The proof is based on the following lemma:

LEMMA 6.5. *For each $1 \leq d \leq k$, there exist positive semi-definite matrices $X = (X_S)$ with nonnegative entries such that $m(X) = O(1)$ and $\ell_{A,B}(X) \geq 1$ for all $A, B \in \mathcal{C}$ satisfying $d \leq |A \setminus B| \leq 2d$.*

The theorem immediately follows from Lemma 6.5. Indeed, we cover the interval $[1, k]$ with a logarithmic number of intervals of the form $[d, 2d]$. For each of them, we apply Lemma 6.5 and take the sum of the resulting matrices.

So, it remains to prove the lemma. Consider the matrices X_S built in the following way. For each S, X_S is a rank-1 matrix with $X_S[\![A,B]\!] = 2^{-n}$ if both $|A \cap S|$ and $|B \cap S|$ lie in the interval $[k/2 - \sqrt{d},\ k/2 + \sqrt{d}]$, and zeroes elsewhere. Using Lemma 2.1, we get that, for all A,

$$m(X) = \Pr_S\Big[\frac{k}{2} - \sqrt{d} \le |S \cap A| \le \frac{k}{2} + \sqrt{d}\Big] = \Theta(\sqrt{d/k}), \tag{6.6}$$

where S is taken uniformly at random from $2^{[n]}$. Fix $A, B \in \mathcal{C}$, and let $\ell = |A \backslash B|$. Assume $d \le \ell \le 2d$. Again, we have $\Pr_S\big[\frac{k-\ell}{2} - \sqrt{d} \le |A \cap B \cap S| \le \frac{k-\ell}{2} + \sqrt{d}\big] = \Omega(\sqrt{d/k})$. Also, provided that the last condition on $A \cap B \cap S$ holds, we get that $\frac{k}{2} - \sqrt{d} \le |A \cap S| < \frac{k}{2}$ with probability $\Omega(1)$, and similarly for $\frac{k}{2} \le |B \cap S| \le \frac{k}{2} + \sqrt{d}$. Thus,

$$\ell_{A,B}(X) = \Omega(\sqrt{d/k}).$$

Combining this with (6.6), and rescaling the matrices X_S, we get the statement of Lemma 6.5. □

7. Further observations

In this section, we prove two additional results about the problems studied in the previous sections. First, we show that many of the above algorithms can be made exact.

Proposition 7.1. *The quantum algorithms for the L_{OR_k}, the $L_{\textsc{exact-half}_k}$ and the $L_{\textsc{majority}_k}$ problems from Theorems 3.1, 5.1 and 6.1 can be made exact without increasing their complexity.*

Proof. We use the same observation as in Iwama *et al.* (2012). Inputs to all these problems are k-subsets of $[n]$. Due to symmetry, the error probability of any of these algorithms is the same on all inputs. Also, for each of the problems, there exists a deterministic procedure that efficiently tests whether a given k-subset A is the true input. Indeed, for the OR function, query the complement of A. For exact-half, cover A by 3 subsets of size $\lfloor k/2 \rfloor$ and query each of them. Similarly for the majority function.

This means that we can apply the exact amplitude amplification algorithm from Brassard *et al.* (2002), and get an exact algorithm with an $O(1)$ multiplicative overhead in complexity. □

Next, we show that the query complexity achieved in the previous sections cannot be obtained by a non-adaptive quantum query algorithm. If h is the OR function, any non-adaptive quantum algorithm requires $\Omega(k)$ queries. This follows from Zalka's result (Zalka 1999) and the fact that unstructured search can be reduced to L_{OR}. For the remaining problems, we obtain the following result:

THEOREM 7.2. *The non-adaptive quantum query complexity of* L_{MAJORITY_k} *and* $L_{\mathrm{EXACT\text{-}HALF}_k}$ *is* $\Omega(\sqrt{k})$.

Note that this result is nearly tight: Using Fourier sampling like in Atıcı & Servedio (2007), it is possible to solve both problems in $\tilde{O}(\sqrt{k})$ quantum queries non-adaptively. Indeed, the Fourier spectrum of the majority and the exact-half functions is concentrated on sets of size roughly $\sqrt{k}$, so, after $O(\sqrt{k}\log k)$ Fourier samples, it is likely to have seen all k relevant variables.

PROOF OF THEOREM 7.2. Essentially, we use the non-adaptive version of the adversary bound from Koiran *et al.* (2010). We give a direct proof, however. Consider a non-adaptive T-query algorithm for one of these problems on n variables. The state of the algorithm before the query is of the form

$$\psi = \sum_{S_1,\ldots,S_T} \alpha_{S_1,\ldots,S_T} |S_1,\ldots,S_T\rangle |\phi_{S_1,\ldots,S_T}\rangle,$$

where S_i are subsets of $[n]$, and $\phi_{S_1,\ldots,S_T}$ are some unit vectors. Assuming $T = o(\sqrt{k})$, we are going to construct two subsets A and B such that $O_A^{\otimes T}\psi$ and $O_B^{\otimes T}\psi$ have large inner product. For the latter, we have

$$\left|\left\langle O_A^{\otimes T}\psi, O_B^{\otimes T}\psi\right\rangle\right| \geq 2\sum |\alpha_{S_1,\ldots,S_T}|^2 - 1, \tag{7.3}$$

where the summation is over all $(S_1,\ldots,S_T)$ such that $f_A(S_i) = f_B(S_i)$ for all $i \in [T]$.

The subsets A and B will be such that $A \cap B = D$ with $|D| = k-1$. Then, $f_A(S) = f_B(S)$ if $|S \cap D| \notin \{\lceil k/2 \rceil - 1, \lceil k/2 \rceil\}$ for both cases of h equal to MAJORITY_k or EXACT-HALF_k.

It is easy to show, using Lemma 2.1, that if D is a $(k-1)$-subset of $[n]$ taken uniformly at random, and n is large enough, then, for any $S \subseteq [n]$, the probability of $|S \cap D| \in \{\lceil k/2 \rceil - 1, \lceil k/2 \rceil\}$ is $O(1/\sqrt{k})$. By the union bound, the probability that $f_A(S_i) = f_B(S_i)$ for all $i \in [T]$ is $1 - o(1)$. By the linearity of expectation, the expectation of the right-hand side of (7.3) is $1 - o(1)$. Hence, there exist A and B such that it is not possible to distinguish $O_A^{\otimes T}\psi$ and $O_B^{\otimes T}\psi$ with error probability less than $1/3$. $\square$

8. Discussion

In this paper, we studied the quantum query complexity of the function L_h, when h is the OR, the exact-half, and the majority function. For the first two functions, we gave optimal algorithms. The algorithms are based on the adversary bound and attain at least quartic improvement in query complexity in comparison with the randomized algorithms when h is the exact-half or the majority function. This shows that the dual adversary bound can be an important tool for quantum learning algorithms.

One apparent open problem is the study of $Q(L_h)$ for other functions h. For instance, can our solution in Section 5 be generalized to the exact-ℓ or the ℓ-threshold functions? For the majority function, there is still an exponential gap between the lower and the upper bounds that we can prove. If the query complexity is logarithmic, we would get an exponential separation using quantum walks. There is already an example of such separation (Childs *et al.* 2003), but the problem studied in the latter paper is not so natural. However, we believe that the complexity is polynomial in k. In this case, we would get an example of a quantum query lower bound outperforming the positive-weighted adversary. There are not so many cases known when a general adversary is strictly better than a positive-weighted adversary (Belovs & Rosmanis 2014; Reichardt & Špalek 2012). Of course, it is also possible to use the polynomial method, as was done for the collision problem in Aaronson & Shi (2004).

Another open problem is to use these ideas in the development of other learning or property testing algorithms. For instance, the combinatorial group testing problem is related to junta testing. Is it possible to use any ideas from the current paper to improve the algorithm in Atıcı & Servedio (2007)?

Acknowledgements

I am grateful to Ashley Montanaro for introducing me to this problem and for helpful advice. I would also like to thank Andris Ambainis, Mihails Belovs, Ansis Rosmanis for useful discussions, and Oded Regev for pointing out the reference Iwama *et al.* (2012) to me. I thank anonymous referees for many useful suggestions on improving the readability of the paper.

The author is supported by Scott Aaronson's Alan T. Waterman Award from the National Science Foundation. This research was performed when the author was at the University of Latvia and was supported by the European Social Fund within the project "Support for Doctoral Studies at University of Latvia" and by ERC Advanced Grant MQC.

A. Basics of representation theory

In this appendix, we formulate the basic results in representation theory of the symmetric group used in Section 4. For general representation theory of finite groups, the reader may refer to Serre (1977) and Curtis & Reiner (1962). For representation theory of the symmetric group, we mostly use Sagan (2001).

An *algebra* A over a field K is a vector space over K that is simultaneously a ring with the identity element. Moreover, the algebra A has to satisfy the following associativity condition: $\alpha(uv) = (\alpha u)v = u(\alpha v)$ for all $\alpha \in K$ and $u, v \in A$.

The only type of algebra we use in the paper is the *group algebra*. Let G be a finite group. The group algebra KG is the vector space over K with the elements of G forming a basis. The ring multiplication operation for the basis elements of KG is inherited from the group G and then uniquely extended by linearity for the remaining elements. That is, $\left(\sum_{g\in G} \alpha_g g\right)\left(\sum_{h\in G} \beta_h h\right) = \sum_{g,h\in G} \alpha_g \beta_h (gh)$.

Assume that A is an algebra over K. A (left) A-module is a vector space M over K such that for all $u \in A$ and $m \in M$, the product um is defined that satisfies the following conditions:

$$u(m+n) = um + un, \quad (u+v)m = um + vm, \quad (uv)m = u(vm),$$
$$em = m, \quad (\alpha u)m = \alpha(um),$$

for all $\alpha \in K$, $u, v \in A$, and $m, n \in M$, and e is the identity element of A. A *submodule* of M is a subspace of M that is closed under multiplication by the elements of A. A module M is called *irreducible* if it does not contain any submodule except for the trivial ones: M itself, and the zero-dimensional subspace $\{0\}$.

Assume that M and N are A-modules. An *A-homomorphism* from M to N is a linear operator $\theta\colon M \to N$ that satisfies $\theta(um) = u\theta(m)$ for all $u \in A$ and $m \in M$. Let $\mathrm{Hom}(M, N)$ denote the linear space of all A-homomorphisms from M to N. If an A-homomorphism θ is also a linear isomorphism, then θ is called an *A-isomorphism*, and M and N are called A-isomorphic.

A *direct sum* $M \oplus N$ of M and N as linear spaces is an A-module with the operation $u(m \oplus n) = um \oplus un$ for all $u \in A$, $m \in M$ and $n \in N$.

A.1. Representations. We only consider $\mathbb{R}G$-modules, where G is a finite group, and $\mathbb{R}$ is the field of real numbers. Such modules are known as (real) *representations*. To define an $\mathbb{R}G$-module M, it suffices to define the products gu, where $g \in G$, and u is a basis element of M. The operation $u \mapsto gu$ is also known as *group action*. We assume that M is equipped with an inner product satisfying $\langle u, v\rangle = \langle gu, gv\rangle$ for all $g \in G$ and $u, v \in M$. Such an inner product can be always constructed (Sagan 2001, Proof of Theorem 1.5.3).

Lemma A.1 (Schur's Lemma, Serre 1977, Section 2.2). *Assume $\theta\colon V \to W$ is an $\mathbb{R}G$-homomorphism between two irreducible $\mathbb{R}G$-modules V and W. Then, $\theta = 0$ if V and W are not isomorphic. Otherwise, θ is uniquely defined up to a scalar multiplier.*

Maschke's theorem (Sagan 2001, Theorem 1.5.3) implies that any $\mathbb{R}G$-module is decomposable into a direct sum of pairwise orthogonal irreducible $\mathbb{R}G$-modules:

$$M = M_1 \oplus M_2 \oplus \cdots \oplus M_m. \tag{A.2}$$

However, this decomposition is not unique.

Let V be an irreducible $\mathbb{R}G$-module. The number of components in (A.2) isomorphic to V is called the *multiplicity* of V in M. Their direct sum is the *canonical submodule* of M associated with V. Both the multiplicity and the canonical submodule do not depend on the decomposition in (A.2) (Serre 1977, Section 2.6).

Let N be a direct sum of ℓ copies of V, and let k be the multiplicity of V in M. Then, Schur's lemma implies that, in a specifically chosen basis, any $\mathbb{R}G$-homomorphism from N to M can be given by $A \otimes I$, where A is an arbitrary $k \times \ell$-matrix and I is the $d \times d$ identity matrix where d is the dimension of V. In particular, the dimension of $\mathrm{Hom}(N, M)$ is $k\ell$.

Assume that M is an $\mathbb{R}G$-module and H is a subgroup of G. Then, M can be also considered as an $\mathbb{R}H$-module. It is called the *restricted* module and is denoted by $M{\downarrow_H}$.

Let G and H be finite groups, M be an $\mathbb{R}G$-module, and N be an $\mathbb{R}H$-module. Then, the tensor product of M and N as vector spaces, $M \otimes N$, is an $\mathbb{R}(G \times H)$-module with the group action defined by $(g, h)(u \otimes v) = (gu) \otimes (hv)$ for all $(g, h) \in G \times H$, $u \in M$ and $v \in N$. This operation is called the *outer tensor product*. The resulting module is irreducible if M and N are irreducible, and every irreducible $\mathbb{R}(G \times H)$-module can be obtained in this way (Serre 1977, Section 3.2).

A.2. Representations of the symmetric group. Throughout this section, X is a finite set of n elements. Let $\mathbb{N}$ denote the set of positive integers. The symmetric group on X is denoted by $\mathbb{S}_X$. It consists of all permutations on X. Clearly, $\mathbb{S}_X$ and $\mathbb{S}_Y$ are isomorphic if $|X| = |Y|$.

A *partition* of n is a sequence $\lambda = (\lambda_i)_{i \in \mathbb{N}}$ of non-increasing non-negative integers that sum up to n, denoted $\lambda \vdash n$. In particular, λ is eventually zero, and its description is usually truncated at the first zero. The *diagram* of λ is defined as $\boldsymbol{\lambda} = \{(i, j) \in \mathbb{N}^2 \mid j \leq \lambda_i\}$.

A *tableau* of shape λ, or λ-*tableau*, is a bijection $t \colon \boldsymbol{\lambda} \to X$. For example, if $\lambda = (3, 1)$, and $X = [4]$,

$$t = \begin{matrix} 1 & 2 & 4 \\ 3 & & \end{matrix}$$

is a tableau with $t(1,1) = 1$, $t(1,2) = 2$, $t(1,3) = 4$, and $t(2,1) = 3$. For $\pi \in \mathbb{S}_X$, the notation πt denotes the composition $\pi \circ t$, which is also a tableau of shape λ. The ith row of t is defined by $R_i(t) = \{t(i,j) \mid (i,j) \in \boldsymbol{\lambda}\}$. The jth column of t is $C_j(t) = \{t(i,j) \mid (i,j) \in \boldsymbol{\lambda}\}$. For each λ-tableau t, we define two subgroups of $\mathbb{S}_X$: $R_t = \prod_i \mathbb{S}_{R_i(t)}$ and $C_t = \prod_j \mathbb{S}_{C_j(t)}$.

The *content* of a function $f\colon X \to \mathbb{N}$ is the sequence $\big(|f^{-1}(i)|\big)$ as i goes through $\mathbb{N}$. Assume $\lambda \vdash n$. A λ-*tabloid* is a function $f\colon X \to \mathbb{N}$ of content λ. The set of all λ-tabloids forms an orthonormal basis of the corresponding *permutation module* M^λ. Let us, for greater clarity, denote the basis element corresponding to f by v_f. The group action on the basis elements is given by $\pi v_f = v_{f \circ \pi^{-1}}$.

For each λ-tableau t, denote $v_t = v_f$, where the λ-tabloid f maps x to $t^{-1}(x)[\![1]\!]$, i.e., to the number of the row of t that contains x. Note that $\pi v_t = v_{\pi t}$. Define the element κ_t of the group algebra $\mathbb{R}\mathbb{S}_X$ by

$$\kappa_t = \sum_{\pi \in C_t} \operatorname{sgn}(\pi)\pi = \prod_j \Bigg(\sum_{\pi \in \mathbb{S}_{C_j(t)}} \operatorname{sgn}(\pi)\pi \Bigg),$$

where $\operatorname{sgn}(\pi)$ denotes the sign of the permutation π. The subspace of M^λ spanned by $\kappa_t v_t$, as t ranges over all λ-tableaux, is an $\mathbb{R}\mathbb{S}_X$-submodule. It is known as the *Specht module* S^λ corresponding to λ (Sagan 2001, Proposition 2.3.5). Each irreducible $\mathbb{R}\mathbb{S}_X$-module is isomorphic to exactly one of the Specht modules (Sagan 2001, Theorem 2.4.6).

Our next aim is to give a description of $\operatorname{Hom}(S^\lambda, M^\mu)$ for partitions λ and μ of n. For that, it is easier to assume that $X = \boldsymbol{\lambda}$. As $\mathbb{S}_X \cong \mathbb{S}_{\boldsymbol{\lambda}}$, this is without loss of generality. In this case, the identity function $\mathrm{id}\colon \boldsymbol{\lambda} \to \boldsymbol{\lambda}$ is a valid λ-tableau. A *generalized tableau of shape* λ is a function $T\colon \boldsymbol{\lambda} \to \mathbb{N}$. The tableau T is called *semi-standard* if $T(i,j+1) \geq T(i,j)$ and $T(i+1,j) > T(i,j)$ for all i,j for which these expressions are defined.

Theorem A.3 (Sagan 2001, Theorem 2.10.1). *For each generalized tableau T of shape λ and content μ, there exists a unique $\mathbb{R}\mathbb{S}_X$-homomorphism $\theta_T\colon M^\lambda \to M^\mu$ satisfying $\theta_T(v_{\mathrm{id}}) = \sum_{\pi\in R_{\mathrm{id}}} \pi v_T$. The corresponding set of restricted homomorphisms $\{\theta_T|_{S^\lambda}\}$, where T runs through the set of all semi-standard generalized tableaux of shape λ and content μ, forms a basis of* $\mathrm{Hom}(S^\lambda, M^\mu)$.

Assume $X = Y \cup Z$ is a partition. Let S^λ, S^μ and S^ν be Specht $\mathbb{S}_X$-, $\mathbb{S}_Y$- and $\mathbb{S}_Z$-modules, respectively. The Littlewood-Richardson rule (Sagan 2001, Section 4.9) gives the multiplicity of $S^\mu \otimes S^\nu$ in $S^\lambda{\downarrow}_{\mathbb{S}_X\times\mathbb{S}_Y}$. The multiplicity is 0 unless $\boldsymbol{\mu} \subseteq \boldsymbol{\lambda}$. Now assume that $\boldsymbol{\mu} \subseteq \boldsymbol{\lambda}$, and consider a function $f\colon \boldsymbol{\lambda} \setminus \boldsymbol{\mu} \to \mathbb{N}$. It is known as a *skew tableau*. A semi-standard skew tableau is defined as for generalized tableaux. The multiplicity of $S^\mu \otimes S^\nu$ in $S^\lambda{\downarrow}_{\mathbb{S}_X\times\mathbb{S}_Y}$ is equal to the number of semi-standard tableaux $f\colon \boldsymbol{\lambda} \setminus \boldsymbol{\mu} \to \mathbb{N}$ of content ν such that the content of the restriction of f onto $\{(i,j) \in \mathbb{N}^2 \mid j \geq a\}$ is non-increasing for any $a \in \mathbb{N}$.

A.3. Johnson association scheme. In this section, we apply the general theory from the previous section to the special case used in Section 4 and prove some results from that section.

Let $N = [n]$. The permutation $\mathbb{S}_N$-module $M(N,k)$ corresponding to a partition $\mu = (n-k, k)$ is known as the *Johnson association scheme*. In this case, we identify a μ-tabloid f with the subset $f^{-1}(2)$. That is, we assume that $M(N,k)$ has the set of all k-subsets of $[n]$ as its orthonormal basis. The tensor product $A \otimes B$ of two disjoint subsets is understood as their union. For example,

$$\big(\{1\} - \{2\}\big) \otimes \big(\{3\} - \{4\}\big) = \{1,3\} - \{1,4\} - \{2,3\} + \{2,4\}$$

is an element of $M(N,2)$ for $n \geq 4$.

Proof of Lemma 4.7. We aim to apply Theorem A.3. Let $\lambda \vdash n$, and $X = \boldsymbol{\lambda}$. If $\lambda_3 > 0$, or if $\lambda_3 = 0$ but $\lambda_2 > k$, then there is no semi-standard generalized tableaux of shape λ and content μ. So, we will further assume that $\lambda_3 = 0$, $\lambda_2 = t \leq k$. In this case, there is unique semi-standard generalized tableau T of shape λ and content μ:

$$T = \begin{matrix} 1 & \dots & 1 & 1 & 1 & \dots & 1 & 2 & \dots & 2 \\ 2 & \dots & 2 & & & & & & & \end{matrix}, \tag{A.4}$$

where there are t occurrences of '2' in the second row, and $k-t$ occurrences in the first row. Hence, $M(N,k) = \bigoplus_{t=0}^{k} S_k(N,t)$.

It is easy to see that the dimension of $M(N,k)$ is $\binom{n}{k}$. As $M(N,k)$ has only one additional irreducible submodule, $S^{(n-k,k)}$, compared to $M(N,k-1)$, the dimension of $S^{(n-k,k)}$ is $\binom{n}{k} - \binom{n}{k-1}$.

By Theorem A.3, $\mathrm{Hom}(S^\lambda, M^\mu)$ is 1-dimensional. Moreover, the only (up to a scalar factor) $\mathbb{RS}_X$-homomorphism $\theta\colon S^\lambda \to M^\mu$ maps $\kappa_{\mathrm{id}} v_{\mathrm{id}}$ into $\kappa_{\mathrm{id}} \sum_{\pi \in R_{\mathrm{id}}} \pi v_T$. Let us analyze the last expression in more detail. The elements of R_{id} permute the elements in the rows of the tableau in (A.4), and the elements of C_{id} permute the elements in its columns. Let $\pi \in R_{\mathrm{id}}$, and $U = T \circ \pi^{-1}$. Then, $U(2,j) = 2$ for all j. If $U(1,j) = 2$ for some $j \le t$, then $\kappa_{\mathrm{id}} v_U = 0$, because, for any $\sigma \in C_{\mathrm{id}}$, σv_U cancels out with $\tau\sigma v_U$, where τ is the transposition exchanging $(1,j)$ and $(2,j)$. Thus, $\kappa_{\mathrm{id}} \sum_{\pi \in R_{\mathrm{id}}} \pi v_T$ is proportional to a linear combination of generalized tableau U of the same form as T, where each of the first t columns of U contain one '1' and one '2', and some of the next $k-t$ columns contain '2'. Moreover, the coefficient of U in this linear combination is 1 if its second row contains even number of '1's, and -1 otherwise.

Let us now translate this to $\mathbb{RS}_N$. Let $a = (a_1, \dots, a_t)$ and $b = (b_1, \dots, b_t)$ be two disjoint sequences of pairwise distinct elements of N. We choose a bijection $t\colon \boldsymbol{\lambda} \to N$ such that $t(1,j) = b_j$ and $t(2,j) = a_j$ for all $j \in [t]$. In other words, we identify the positions in the tableau with integers in N. In our interpretation, a generalized tableau U corresponds to the set of positions labeled by '2'. Thus, if we apply the bijection t to the homomorphism θ, we get that the only $\mathbb{RS}_N$-homomorphism from S^λ to M^μ maps the vector

$$\big(\{a_1\} - \{b_1\}\big) \otimes \big(\{a_2\} - \{b_2\}\big) \cdots \otimes \big(\{a_t\} - \{b_t\}\big)$$

(corresponding to $\kappa_{\mathrm{id}} v_{\mathrm{id}}$) into the vector

$$\big(\{a_1\} - \{b_1\}\big) \otimes \cdots \otimes \big(\{a_t\} - \{b_t\}\big) \otimes \Big(\sum_{A \subseteq N \setminus \{a_1,\dots,a_t,b_1,\dots,b_t\}\colon |A| = k-t} A \Big)$$

(corresponding to $\kappa_{\mathrm{id}} \sum_{\pi \in R_{\mathrm{id}}} \pi v_T$). $\square$

Proof of (4.11). Let $N = N_0 \cup N_1$ be a partition, and let $S_k(N,t)$ be the unique copy of $S^{(n-t,t)}$ in $M(N,k)$. We aim to apply the Littlewood-Richardson rule in order to get the decomposition of $S_k(N,t)\downarrow_{\mathbb{S}_{N_0}\times\mathbb{S}_{N_1}}$ into irreducible submodules $S^\mu \otimes S^\nu$. As before, the multiplicity is zero if $\mu_3 > 0$, $\nu_3 > 0$, $\mu_2 > k$, or $\nu_2 > k$. Thus, let us assume $\mu_3 = \nu_3 = 0$, and $\mu_2 = t_0$, $\nu_2 = t_1$ satisfy $t_0, t_1 \leq k$. Thus, if there is any skew tableau satisfying the conditions of the Littlewood-Richardson rule, it must have the form

$$\begin{array}{cccccccccc} * & \dots & * & * & \dots & * & \dots & 1 & \dots & 1 \\ * & \dots & * & 1 & \dots & 2 & & & & \end{array},$$

where the $*$ stand for the elements of $\boldsymbol{\mu}$. (The crucial observation here is that the right-most element of the first row must be equal to 1.) That is, the inequality $t_1 + t_2 \leq t$ must hold, and in this case the multiplicity is 1. □

References

Scott Aaronson & Yaoyun Shi (2004). Quantum Lower Bounds for the Collision and the Element Distinctness Problems. *Journal of the ACM* **51**(4), 595–605.

Andris Ambainis (2002). Quantum Lower Bounds by Quantum Arguments. *Journal of Computer and System Sciences* **64**(4), 750–767.

Andris Ambainis, Kazuo Iwama, Akinori Kawachi, Hiroyuki Masuda, Raymond H Putra & Shigeru Yamashita (2004). Quantum identification of boolean oracles. In *Proceedings of the 21st Symposium on Theoretical Aspects of Computer Science*, volume 2996 of *Lecture Notes in Computer Science*, 105–116. Springer-Verlag.

Andris Ambainis & Ashley Montanaro (2014). Quantum algorithms for search with wildcards and combinatorial group testing. *Quantum Information & Computation* **14**(5&6), 439–453.

Dana Angluin (1988). Queries and concept learning. *Machine learning* **2**(4), 319–342.

Alp Atici & Rocco A. Servedio (2005). Improved bounds on quantum learning algorithms. *Quantum Information Processing* **4**(5), 355–386.

Alp Atici & Rocco A. Servedio (2007). Quantum algorithms for learning and testing juntas. *Quantum Information Processing* **6**(5), 323–348.

Aleksandrs Belovs (2012a). Learning-graph-based Quantum Algorithm for k-distinctness. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, 207–216.

Aleksandrs Belovs (2012b). Span programs for functions with constant-sized 1-certificates. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, 77–84.

Aleksandrs Belovs & Ben W. Reichardt (2012). Span programs and quantum algorithms for st-connectivity and claw detection. In *Proceedings of the 20th Annual European Symposium on Algorithms*, volume 7501 of *Lecture Notes in Computer Science*, 193–204. Springer-Verlag.

Aleksandrs Belovs & Ansis Rosmanis (2014). On the Power of Non-Adaptive Learning Graphs. *Computational Complexity* **23**(2), 323–354.

Ethan Bernstein & Umesh Vazirani (1997). Quantum complexity theory. *SIAM Journal on Computing* **26**(5), 1411–1473.

Eric Blais (2009). Testing juntas nearly optimally. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, 151–158.

Stephen Boyd & Lieven Vandenberghe (2004). *Convex optimization.* Cambridge University Press.

Gilles Brassard, Peter Høyer, Michele Mosca & Alain Tapp (2002). Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, 53–74.

Nader H. Bshouty, Richard Cleve, Ricard Gavaldà, Sampath Kannan & Christino Tamon (1996). Oracles and queries that are sufficient for exact learning. *Journal of Computer and System Sciences* **52**(3), 421–433.

Nader H. Bshouty & Jeffrey C. Jackson (1998). Learning DNF over the uniform distribution using a quantum example oracle. *SIAM Journal on Computing* **28**(3), 1136–1153.

Harry Buhrman & Ronald de Wolf (2002). Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science* **288**, 21–43.

Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann & Daniel A. Spielman (2003). Exponential algorithmic speedup by a quantum walk. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, 59–68.

Andrew M. Childs, Robin Kothari, Māris Ozols & Martin Rötteler (2013). Easy and hard functions for the Boolean hidden shift problem. In *Proceedings of the 8th Conference on the Theory of Quantum Computation, Communication and Cryptography*, volume 22 of *Leibniz International Proceedings in Informatics*, 50–79. Schloss Dagstuhl.

Charles W. Curtis & Irving Reiner (1962). *Representation theory of finite groups and associative algebras*. American Mathematical Society.

Wim van Dam (1998). Quantum oracle interrogation: Getting all information for almost half the price. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, 362–367.

Ding Zhu Du & Frank Hwang (1993). *Combinatorial group testing and its applications*, volume 3 of *Series on Applied Mathematics*. World Scientific.

Mark Ettinger, Peter Høyer & Emanuel Knill (2004). The quantum query complexity of the hidden subgroup problem is polynomial. *Information Processing Letters* **91**(1), 43–48.

Paul Hausladen & William K. Wootters (1994). A pretty good measurement for distinguishing quantum states. *Journal of Modern Optics* **41**(12), 2385–2390.

Peter Høyer, Troy Lee & Robert Špalek (2007). Negative weights make adversaries stronger. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, 526–535.

Kazuo Iwama, Harumichi Nishimura, Rudy Raymond & Junichi Teruyama (2012). Quantum counterfeit coin problems. *Theoretical Computer Science* **456**, 51–64.

Pascal Koiran, Jürgen Landes, Natacha Portier & Penghui Yao (2010). Adversary lower bounds for nonadaptive quantum algorithms. *Journal of Computer and System Sciences* **76**(5), 347–355.

Robin Kothari (2014). An optimal quantum algorithm for the oracle identification problem. In *Proceedings of the 31st Symposium on Theoretical Aspects of Computer Science*, volume 25 of *Leibniz International Proceedings in Informatics*, 482–493. Schloss Dagstuhl.

Ilia Krasikov & Simon Litsyn (2001). Survey of binary Krawtchouk polynomials. In *Codes and association schemes*, volume 56 of *DIMACS series in Discrete Mathematics and Theoretical Computer Science*, 199–212. American Mathematical Society.

Troy Lee, Frédéric Magniez & Miklos Santha (2013). Improved quantum query algorithms for triangle finding and associativity testing. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1486–1502.

Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek & Mario Szegedy (2011). Quantum query complexity of state conversion. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, 344–353.

Ashley Montanaro (2010). Nonadaptive quantum query complexity. *Information Processing Letters* **110**(24), 1110–1113.

Ben W. Reichardt (2009). Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, 544–551.

Ben W. Reichardt & Robert Špalek (2012). Span-program-based quantum algorithm for evaluating formulas. *Theory of Computing* **8**, 291–319.

Bruce E. Sagan (2001). *The symmetric group: representations, combinatorial algorithms, and symmetric functions*, volume 203 of *Graduate Texts in Mathematics*. Springer-Verlag.

Jean-Pierre Serre (1977). *Linear Representations of Finite Groups*, volume 42 of *Graduate Texts in Mathematics*. Springer-Verlag.

Rocco A. Servedio & Steven J Gortler (2004). Equivalences and separations between quantum and classical learnability. *SIAM Journal on Computing* **33**(5), 1067–1092.

Robert Špalek & Mario Szegedy (2006). All Quantum Adversary Methods are Equivalent. *Theory of Computing* **2**, 1–18.

Gabor Szegő (1975). *Orthogonal polynomials*, volume 23 of *AMS Colloquium Publications*. American Mathematical Society.

Christof Zalka (1999). Grover's quantum searching algorithm is optimal. *Physical Review A* **60**(4), 2746.

Bohua Zhan, Shelby Kimmel & Avinatan Hassidim (2012). Superpolynomial quantum speed-ups for Boolean evaluation trees with hidden structure. In *Proceedings of the 3rd Innovations in Theoretical Computer Science conference*, 249–265. ACM Press.

Manuscript received 9 June 2014.

Aleksandrs Belovs
Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology,
Cambridge, MA 02139, USA
abelov@csail.mit.edu
http://people.csail.mit.edu/abelov/