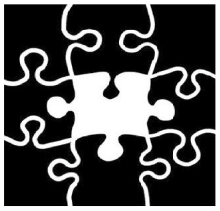
Quantum Algorithms and Learning Theory

Srinivasan Arunachalam

Quantum Algorithms and Learning Theory

ILLC Dissertation Series DS-2018-08



Institute for Logic, Language and Computation

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 107
1098 XG Amsterdam
phone: +31-20-525 6051
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

The investigations were supported by the ERC Consolidator Grant QPROGRESS.

Printed and bound by Ipskamp Drukkers.

ISBN: 978-94-028-0984-8

Quantum Algorithms and Learning Theory

Academisch Proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K.I.J. Maex
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in de Agnietenkapel
op woensdag 25 april 2018, te 14.00 uur

door

Srinivasan Arunachalam

geboren te Bangalore, India

Promotiecommisie

Promotores:	Prof. dr. R. M. de Wolf	Universiteit van Amsterdam
	Prof. dr. H. M. Buhrman	Universiteit van Amsterdam
Overige leden:	Prof. dr. E. M. Opdam	Universiteit van Amsterdam
	Prof. dr. C. J. M. Schoutens	Universiteit van Amsterdam
	Prof. dr. P. D. Grünwald	Universiteit Leiden
	Dr. M. Ozols	Universiteit van Amsterdam
	Dr. A. Montanaro	University of Bristol, UK

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

This thesis is based on the following papers. For the first five papers, the authors are ordered alphabetically and the co-authorship is shared equally. For the sixth paper, the authors are ordered based on their contribution.

1. [AW17c] Srinivasan Arunachalam and Ronald de Wolf. Optimizing the Number of Gates in Quantum Search. In *Quantum Information & Computation, 17(3&4):251-261, 2017.*

2. [ABP18] Srinivasan Arunachalam, Jop Briët, and Carlos Palazuelos. Quantum query algorithms are completely bounded forms. In *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science (ITCS), pages 3:1-3:21, 2018.*

3. [AW17a] Srinivasan Arunachalam and Ronald de Wolf. Guest column: A survey of quantum learning theory. In *SIGACT News, 48(2):41-67, 2017.*

4. [AW17b] Srinivasan Arunachalam and Ronald de Wolf. Optimal quantum sample complexity of learning algorithms. In *32nd Computational Complexity Conference (CCC), pages 25:1-25:31, 2017.*

5. [ACLW18] Srinivasan Arunachalam, Sourav Chakraborty, Troy Lee, and Ronald de Wolf. Two new results on quantum exact learning. Manuscript.

6. [GAW17] András Gilyén, Srinivasan Arunachalam, and Nathan Wiebe. Optimizing quantum optimization algorithms via faster quantum gradient computation. Preprint available at arXiv:1711.00465 [quant-ph].

In the course of his PhD, the author has additionally (co-)authored the following articles that are not included in this thesis (most of the work in these projects was done for his Master's degree).

1. [AMR17] Srinivasan Arunachalam, Abel Molina, and Vincent Russo. Quantum hedging in two-round prover-verifier interactions. In *12th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2017).*

2. [AGJO+15] Srinivasan Arunachalam, Vlad Gheorghiu, Tomas Jochym-O' Connor, Michele Mosca, and Priyaa Varshinee Srinivasan. On the robustness of bucket brigade quantum RAM. In *10th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2015).* Also in *New Journal of Physics, 17(12): 123010, 2015.*

3. [AJR15] Srinivasan Arunachalam, Nathaniel Johnston, and Vincent Russo Is absolute separability determined by the partial transpose? In *Quantum Information & Computation, 15(7&8):694-720, 2015.*

Contents

Acknowledgments

First and foremost, I would like to thank my advisor Ronald de Wolf, without whom this thesis would definitely have not been possible. Apart from his patient supervision and beautiful insights into various problems that we worked on, most importantly (and I can't stress this enough) he had faith in me. The first two years of my PhD were slow, depressing, felt never-ending and needless to say, scientifically unproductive. When I thought it was time to quit, Ronald continued encouraging me, reiterating that things were going fine. Since you are reading this, I guess Ronald was right after all and his faith in me was justified maybe. I really thank him for this. There were two memorable moments in these four years. First, when Ronald went nuts when I was sloppy (and didn't know!) when using the Kleene star notation[1] and this emphasized the importance of rigor in his style of research. Second, when Ronald finally agreed to work on quantum learning theory with me and those collaborations contributed to a major part of this thesis. Overall, it has been an amazing learning experience being a student and collaborator of Ronald. Thanks for the opportunity.

I am grateful to the members of my PhD committee, Harry Buhrman, Eric Opdam, Kareljan Schoutens, Peter Grünwald, Maris Ozols and Ashley Montanaro, for agreeing to be part of my committee and for helpful comments on this thesis.

This thesis is easily $(1-\varepsilon)$-far from being a single person's work. Every paper included here came with some wonderful collaborators: Jop Briët, Sourav Chakraborty, András Gilyén, Troy Lee, Carlos Palazuelos, Nathan Wiebe and Ronald de Wolf. It has been a truly humbling experience working with these incredibly brilliant researchers. In particular, I would like to thank Jop. He got me involved in a project on quantum query complexity hoping to use my knowledge of the subject but he ended up knowing more about it! His patient and quiet approach to problem-solving and beautiful insights into problems still amazes me. I would also like to thank Carlos and Jop for vastly improving my

[1] Any future PhD student of Ronald reading this, you better know what the Kleene star notation is!

knowledge on operator space theory. Apart from collaborations which resulted in papers, Ronald frequently had postdocs and visitors, from whom I have learnt a lot. In particular, I would like to thank Henry Yuen: his summer visits to CWI were the best time to work on hard problems. I also thank Ralph Bottesch, Sourav Chakraborty, Robin Kothari, Nishant Mehta and Penghui Yao, Henry Yuen for vastly improving my knowledge in communication complexity, query complexity, Fourier analysis and learning theory.

There were other PhD students at CWI who may not have solved my research questions, but were there to chat when I wanted to. In particular, Jeroen was always there to talk to about PhD work and life. I also had lot of fun talking to and hanging out with Teresa (thanks for dragging me out of my office on many occasions), Tom (thanks for writing my samenvatting and importantly ensuring that we won the Foosball tournament!), Gabriele, Chris, Lars, Tom (the philosopher), Deba, Florian, Koen and Joris (who still haven't kicked me out of my office for making fun of physicists), Isabella and Jeroen (working on the weekends wouldn't have been "fun" without you guys), Yfke, Jan, András, Farrokh, Joran and Alvaro.

My stay in Amsterdam wouldn't have been fun if not for some interesting flatmates. In particular, I thank Chetan and Gorcia (and their daughter Anya), Sudheer, Sirshendu and Soumen for all the fun times. Thanks for the awesome food and fun conversations. A special mention to Sirshendu, my interest in chess and trekking might not have started if not for conversations with you. I had some amazing friends not living in Amsterdam who still had a major impact on my PhD life. Firstly, I would like to thank Preetham and TR, in particular their enthusiasm to travel. They dragged me all the way from Russia to Kilimanjaro and the times spent with them and the discussions we had were awesome. In the same light, I also thank Ananya for trying to drag me outside Netherlands (and not succeeding) and all the fun conversations. Finally, I would like to thank Akash and Swati. They have been friends since the Canada days and in these four years, the number of Skype and Gtalk conversations with them are uncountable, talking to them always turned a frown into a smile.

Finally, I would like to dedicate this thesis to my parents. They gave up a safe, comfortable and peaceful life in Singapore and moved to Bangalore, just to ensure that my quality of studies was better. Through this thesis, I hope they realize that their sacrifice was well worth it. The sacrifices they have made for me doesn't end there, they have always put my well-being in front of theirs trying to ensure that I live a happy and comfortable live. They have been supportive of everything that I have wanted to do. It is simply impossible to express how thankful I am for everything that they have done for me.

Srinivasan Arunachalam
February 2018, Amsterdam

Chapter 1

Overview

Contents

1.1 Introduction

The field of quantum computation was initiated in the early 1980s by Richard Feynman [Fey82, Fey85], Yuri Manin [Man80, Man99], Paul Benioff [Ben82] and David Deutsch [Deu85] when they realized that a quantum computer, whose working is based on the laws of quantum mechanics, could possibly be more efficient than classical Turing machines for simulating quantum systems. One of the reasons why quantum computers are believed to be more powerful than their classical analogue is the peculiar feature called *superposition*, which is allowed by quantum mechanics but not classical physics. A fundamental building block of a classical computer is a bit, which is either in the state 0 *or* 1. The building block of a quantum computer is a quantum bit (also called *qubit*) that can be in a superposition, i.e., the state of a qubit can be simultaneously 0 *and* 1, each associated with an amplitude.

Quantum computation gained significant attention after Shor's polynomial-time quantum algorithm [Sho97] in 1994 for factoring integers and computing discrete logarithms (which break much of today's public-key cryptography) and

Grover's quantum algorithm [Gro96] in 1996, which searches for a marked item in an unstructured dataset quadratically faster than every possible classical search algorithm. Quantum computing has since blossomed into a major field at the intersection of physics, mathematics, and computer science. The past two decades have seen much research in trying to understand what are the tasks for which quantum provides an advantage.

In the first part of this thesis, our focus will be on quantum algorithms. In particular, we will consider two natural complexity measures used often to understand and compare classical and quantum algorithms, *gate complexity* and *query complexity*. Gate complexity deals with the number of gates used in the implementation of algorithms. In the classical setting we are referring to Boolean logic gates (such as AND, OR, NAND gates) and in the quantum setting we are referring to elementary quantum gates (such as Hadamard, CNOT, single-qubit Pauli gates). However, proving lower bounds on the number of gates required to solve certain problems is extremely hard. This prompts the question, is there a simpler measure that allows us to understand the power of quantum computers? Query complexity is one such information-theoretic measure that is often used to give unconditional separations between quantum and classical computing.

In the second part of this thesis, we discuss another field that was also conceptualized in the early 1980s, *computational learning theory*. Leslie Valiant's seminal paper "A Theory of the Learnable" [Val84] laid the foundation to computational learning theory, which has since evolved into a field that is used to mathematically understand and analyze machine learning algorithms. In the last decade, with the explosion of data and computing power, heuristic approaches such as deep learning have gained prominence. Deep learning is extremely good in practice for natural language processing, speech recognition, computer vision, even the games of Go and chess. Alongside the boom in classical machine learning algorithms, the last few years have seen an increase in the interest in *quantum machine learning*, an interdisciplinary area that uses the powers of quantum physics to improve machine learning algorithms. Given the practical relevance of machine learning, it is believed that quantum machine learning algorithms implemented on small-scale quantum devices may become one of the first interesting and practically relevant application of quantum computers.

The main motivation behind the research in this thesis is to broadly understand *query and gate complexity* of quantum algorithms for certain problems and the *sample and query complexity* of quantum machine learning algorithms. In this chapter, we briefly describe the model of query complexity and learning theory and preview our contributions in this thesis.

1.2 Quantum algorithms

1.2.1 A brief introduction to the query model

The primary object of study in the first part of this thesis is the following problem. Let f be a *known* Boolean function $f : \{0,1\}^N \to \{0,1\}$, i.e., f maps n-bit strings to either 0 or 1. Suppose an adversary picks an *unknown* $x \in \{0,1\}^N$. Our goal is to compute $f(x)$. Clearly computing $f(x)$ is not possible without further information. Suppose we can *query* the adversary by asking questions *only* of the form "what is x_i", i.e., the ith bit of x (for some choice of $i \in \{1,\ldots,N\}$ of our choice) to which the adversary responds with x_i. After the query, we are allowed to perform arbitrary operations. We repeat this question-response process a few times before outputting a bit b. In the *deterministic* model, the output bit b should equal $f(x)$ with certainty. In the *randomized* model (where we allow randomness in the algorithm), the output bit b should equal $f(x)$ with probability at least $2/3$ (where the probability is taken over the internal randomness of the algorithm). There are other query models which we do not discuss in this thesis, such as the non-deterministic model, the unbounded-error model, query complexity computing in expectation, the non-adaptive query model. The question we are interested in is, *how many queries* to the adversary suffice to compute $f(x)$ in the respective query models? Clearly N questions suffice, because we could simply ask the adversary the following N questions: "what is x_1", "what is x_2", "what is x_3",...,"what is x_N" and learn x completely. Since we now know f and x, we can compute $f(x)$. Can we make *fewer* than N queries and still learn $f(x)$? *Query complexity* tries to understand this question for different Boolean functions f under different query models. Note that in query complexity we are not interested in the number of gates used in between queries. *Gate complexity* tries to understand the number of gates used by the entire algorithm before it can decide $f(x)$.

In this thesis we will be interested in the *quantum* query model. In the quantum query model, we replace the classical questions "what is x_i" by a quantum superposition of questions. A quantum superposition of questions is commonly referred to as a *quantum query*. The central question in the field of quantum query complexity is to understand to what extent quantum queries can reduce the query complexity of certain Boolean functions. Although constructing query-efficient quantum algorithms is the goal, it is also desirable that the number of quantum gates used in these query algorithms is not much more than their query complexity.

The beauty of the quantum query model is that *almost all* existing quantum algorithms work in the query model. In fact, the first few breakthroughs in quantum algorithms, which piqued the interest of many researchers, were in the quantum query model (see Deutsch-Josza algorithm [Deu85], Simon's al-

gorithm [Sim97], Shor's factoring algorithm [Sho97][1] and Grover's search algorithm [Gro96]). The query model captures most problems for which one can *provably* show a polynomial or even exponential speed-up in the quantum setting.

1.2.2 Our contributions to quantum algorithms

In this thesis we present three contributions to quantum algorithms. We describe these results in the following three paragraphs.

Improving the gate complexity of Grover's search algorithm. As we described earlier, one of the first successes of quantum computing is Grover's search algorithm [Gro96]. Consider the following problem: suppose we have an N-element unstructured database and we are promised that there is a unique item in the database that is "marked". The goal is to find the marked item. To solve the problem, we are allowed to make "database queries", which tell us if a single item is marked or not and our goal is to find the marked item making as *few* database queries as possible. Classically, it is not hard to see that in the worst case, we need to essentially make N database queries in order to find the marked item.

Grover [Gro96] constructed a quantum algorithm that finds the marked item using $O(\sqrt{N})$ *quantum* database queries and his algorithm involved $O(\sqrt{N}\log N)$ other elementary gates. This quantum algorithm already allows us to quadratically improve *almost all* classical search subroutines. In fact many quantum algorithms use Grover's search algorithm as a subroutine to improve classical algorithms.

Is Grover's algorithm optimal? Could there be a better quantum search algorithm? It was shown that $\Omega(\sqrt{N})$ quantum queries are necessary [BBBV97] to solve the search problem, so Grover's algorithm cannot be improved in terms of queries. But what about the number of elementary gates? Can this be improved? In Chapter 3 we give a positive answer to this question. We construct a new search algorithm whose gate complexity is essentially $O(\sqrt{N})$,[2] while preserving the query complexity of Grover's algorithm. Although our improvement might seem not so significant since we essentially only remove a logarithmic factor, it is interesting that after two decades of research, the basic quantum search algorithm can still be improved in some ways!

New characterization of quantum query algorithms. Moving away from constructing better quantum query algorithms, it is also important to understand

[1] Although Shor's algorithm is technically not a query algorithm, the heart of Shor's algorithm is a quantum query algorithm that solves the period-finding problem exponentially faster than every classical algorithm.

[2] Strictly speaking, our gate complexity is $O(\sqrt{N}\log(\log^* N))$. However, $\log(\log^* N)$ for all "practical" purposes is a constant since $\log(\log^*(2^{10000})) \leq 3$.

their *limitations*. The flip-side of obtaining new algorithms is showing query *lower bounds*, i.e., showing that *every* quantum algorithm needs to make at least a certain number of queries before solving a problem. Proving such lower bounds seems significantly more challenging than constructing specific quantum algorithms. In this direction, there are two famous techniques to give query lower bounds: the *polynomial method* [BBC+01] and the *adversary method* [Amb00]. The former method uses that the approximate polynomial degree of a Boolean function (an algebraic parameter that we define in the next chapter) lower bounds quantum query complexity and the latter method uses properties of the spectral norm of a so-called "adversary matrix" to give lower bounds on quantum query complexity.

The polynomial method was initially used to prove lower bounds for the search problem and other symmetric functions [BBC+01], the collision problem and element distinctness [AS04]. In the last decade, the adversary method has been the favoured lower bounding technique primarily because the "negative-weight" adversary method (which generalized the positive-weight adversary method introduced by Ambainis [Amb02]) was shown to *characterize* quantum query complexity [HLŠ07, Rei09, Rei11, LMR+11], i.e., upper bounds on the negative-weight adversary method also gave upper bounds on quantum query complexity. However, using the adversary method to prove *good* quantum query bounds appears to be hard in general!

In this thesis we consider if the polynomial method admits such a converse. If this were true, this would imply a succinct *characterization* of quantum algorithms in terms of polynomials and also give an alternate method (to the adversary method) to showing quantum query lower bounds. However, Ambainis [Amb06] already answered this question in the negative. This leaves open: does there exist a (simple) refinement of approximate polynomial degree that characterizes quantum query complexity? This was explicitly raised in recent works of Aaronson and others [AA15, AAI+16]. In Chapter 4 we give a positive answer to this question. We refine the polynomial method and obtain a new notion of polynomial degree, called *completely bounded approximate degree*, that *equals* quantum query complexity. Our new characterization of quantum algorithms in terms of polynomials not only refines the well-known polynomial method, but it also gives a new method for showing upper and lower bounds on quantum query complexity.

Better algorithm to compute the gradient of a multivariate function. Optimization is a fundamentally important task that touches on virtually every area of science. Naïvely, since Grover's search algorithm [Gro96] quadratically improves upon the classical algorithm for searching in a database, we can simply use it to speed up all discrete optimization algorithms which involve searching for a solution among a set of unstructured candidate solutions. However, applying non-Grover techniques to real-word optimization problems has proven challenging, because generic problems usually fail to satisfy the delicate requirements of

these advanced quantum techniques. There have been many works on improving specific optimization techniques such as the Monte Carlo method [Mon15], quantum adiabatic optimization [KN98, FGGS00, FGG14], improving optimization algorithms for the traveling salesman problem [HP00], Boolean satisfiability [Aru14], least-squares fitting [WBL12] and so on.

In this thesis we consider a generic framework of gradient-based optimization, ubiquitously used in continuous-variable optimization. For $f : \mathbb{R}^d \to \mathbb{R}$, consider the following optimization problem: $\text{OPT} = \min\{f(x) : x \in \mathbb{R}^d\}$. One generic technique used often to compute OPT is the gradient-descent algorithm. This begins with an arbitrary $\boldsymbol{x} \in \mathbb{R}^d$, computes the gradient of f at $\boldsymbol{x}$ (denoted $\nabla f(\boldsymbol{x})$) and moves to a point $\boldsymbol{x'}$ in the direction of $-\nabla f(\boldsymbol{x})$. This process is repeated a few times before the algorithm hopefully obtains a good approximation of OPT. Given the simplicity and generality of the algorithm, gradient-based methods are used often in machine learning algorithms.

In Chapter 5 we develop a quantum algorithm that calculates the gradient of $f : \mathbb{R}^d \to \mathbb{R}$ quadratically faster than classical gradient computation algorithms. For a class of *smooth* functions, our quantum algorithm provides an approximation of the gradient vector ∇f with quadratically better dependence on the evaluation accuracy of f. To be precise, we show that in order to obtain an ε-coordinate-wise approximation of the d-dimensional gradient vector ∇f, it suffices to make $\widetilde{O}(\sqrt{d}/\varepsilon)$ queries to the oracle encoding f. Furthermore, we show that most functions arising from quantum optimization algorithms satisfy the smoothness condition. Using this, we obtain a quadratic quantum improvement in the complexity of most gradient-based optimization algorithms. In particular, our quantum improvement quadratically improves the complexity of most machine learning algorithms that rely on gradient-based methods. We also show that our quantum algorithm for gradient calculation is optimal (for a class of smooth functions).

1.3 Learning in a quantum world

1.3.1 A brief introduction

In the second part of this thesis we discuss the theoretical aspects of machine learning. We first discuss the classical learning model, before discussing its quantum generalization. A *concept class* $\mathcal{C}$ is a collection of n-bit Boolean functions $\{c_1, \ldots, c_m\}$ where $c_i : \{0,1\}^n \to \{0,1\}$. Suppose an adversary picks an *unknown* $c_i \in \mathcal{C}$. The goal of a learner is to learn the unknown target concept, either exactly or approximately. There are two models of learning which we discuss in this thesis.

1. *Exact learning*: This is similar to the model of query complexity. In exact learning, the learner can actively query the adversary by asking it questions

of the form "what is $c_i(x)$" for some x chosen by the learner. Such a query is referred to as a *membership query*. This process repeats until the learner can identify the target concept. In exact learning, we are concerned with the number of *membership queries* that suffice to learn c_i? Clearly 2^n queries suffice because the learner could query c_i on every $x \in \{0,1\}^n$ and learn the truth table of c_i. In exact learning, we would like to understanding if *fewer* membership queries suffice to learn the "hardest" function in the concept class $\mathcal{C}$?

2. *PAC learning*: The Probably Approximately Correct (PAC) model of learning is a well-known *passive* learning model. In PAC learning, there is an unknown distribution D on the set of n-bit strings. Here the learner can no longer query c_i on an arbitrary input x of its choice, instead the learner obtains *labelled examples* $(x, c_i(x))$ where x is drawn according to the unknown distribution D.[3] In PAC learning, it is not even clear that 2^n labelled examples suffice to exactly learn c_i because D is an arbitrary unknown distribution! Instead of identifying the target concept, in the PAC model the learner needs to output an *hypothesis* h which is "close" to c_i under the distribution D, i.e., $\Pr_{x\sim D}[h(x) \neq c_i(x)] \leq \varepsilon$ (for some $\varepsilon > 0$). In PAC learning, we are concerned with two measures of complexity, the number of labelled examples and the time taken to learn the "hardest" function in the concept class $\mathcal{C}$. They are many variants of PAC learning which we discuss in subsequent chapters.

The quantum generalization. The quantum generalization of exact learning is directly motivated by quantum query complexity. Instead of classical queries, a quantum learner can make quantum queries. The central question is to understand if fewer quantum queries suffice to exactly learn a concept class $\mathcal{C}$. Indeed, using results from query complexity one can construct concept classes for which quantum queries can provide an advantage. For example, consider the concept class on N bits,[4] $\mathcal{C} = \{(10\cdots0), (010\cdots0), \ldots, (0\cdots01)\}$. Observe that identifying an unknown concept $c \in \mathcal{C}$ is equivalent to identifying a unique marked item in an N-element database. So, we can use Grover's search algorithm to identify an unknown $c \in \mathcal{C}$ using $O(\sqrt{N})$ quantum queries.

Given the advantage of quantum queries in the model of query complexity, one important goal in quantum learning theory is to understand if fewer quantum queries suffice to learn a concept class $\mathcal{C}$? For a concept class $\mathcal{C}$ on N bits, suppose $D(\mathcal{C})$ and $Q(\mathcal{C})$ are the classical and quantum membership query

[3] The inability of the learner to query c_i on an x of its choice is why PAC learning is a passive learning model. In contrast, the exact learning model is referred as an *active* learning model.

[4] Observe that concepts $c : \{0,1\}^n \to \{0,1\}$ can be identified with their $N = 2^n$-bit truth tables $c \in \{0,1\}^N$. So a concept class $\mathcal{C} \subseteq \{c : \{0,1\}^n \to \{0,1\}\}$ can also be viewed as a subset of N-bit strings.

complexities of $\mathcal{C}$ respectively. Then Servedio and Gortler [SG04] showed that $D(\mathcal{C}) \leq O(Q(\mathcal{C})^3 \log N)$.

The quantum generalization of PAC learning was introduced by Bshouty and Jackson [BJ99], who defined a quantum example as thestate

$$\sum_x \sqrt{D(x)}|x, c_i(x)\rangle.$$

While it is not always realistic to assume access to such (fragile) quantum states, one can certainly envision scenarios where the data is provided by a coherent quantum process. Such a quantum example is the natural quantum generalization of a classical labelled example. In order to see this, suppose a learner chooses to measure this quantum example state. Then the learner would obtain an $(x, c_i(x))$ pair with probability $D(x)$, just like a classical PAC learner. However, Bshouty and Jackson [BJ99] exhibited a concept class and distribution D, under which quantum examples gave a large advantage in learning the concept class compared to classical examples.

When can we expect quantum examples to help in PAC learning? Are they useful for every concept class $\mathcal{C}$ and distribution D? This leads to another important question, understanding the limitations of quantum examples for PAC learning. Classically, it is well-known that the number of classical examples necessary and sufficient for PAC learning a concept class $\mathcal{C}$ is given by a combinatorial parameter called the VC dimension of $\mathcal{C}$ (denoted VC-dim($\mathcal{C}$)), named after Vapnik and Chervonenkis [VC71]. Atıcı and Servedio [AS05] showed that the quantum sample complexity of PAC learning is at least $\Omega(\sqrt{\text{VC-dim}(\mathcal{C})}/\varepsilon)$, which leaves room to show that quantum examples are possibly advantageous for all concept classes $\mathcal{C}$ and distributions D.

1.3.2 Our contributions to learning theory

In this thesis we present two contributions to quantum learning theory. We describe these contributions in the following two paragraphs.

Survey on quantum learning theory. In recent times quantum machine learning has been well-served by a number of survey papers [SSP15, AAD+15, BWP+17, CHI+17, DB17] and even a book [Wit14]. In contrast, there has not been much work on understanding quantum learning from a theoretical perspective. In Chapter 6, we include a survey of quantum learning theory, which we were invited to write for the SIGACT complexity theory column [AW17a]. We focus on the theoretical side of quantum machine learning: quantum learning theory. We describe the main results known for three models of learning, using classical as well as quantum data: exact learning from membership queries, the probably approximately correct (PAC) learning model and the agnostic learning model, which is a more realistic and flexible version of the PAC learning

model. Apart from information-theoretic results, we also survey results on the time complexity of learning from membership queries and learning in the PAC and agnostic models.

Optimal sample complexity of learning algorithms. We saw earlier that the number of labelled examples necessary and sufficient for a good classical PAC learner trying to learn $\mathcal{C}$ is *characterized* by the VC dimension of $\mathcal{C}$. What is the sample complexity when a quantum learner is given access to quantum examples, in terms of VC-dim($\mathcal{C}$)? Since a quantum learner could simply measure the quantum example and obtain a labelled example like the classical PAC learner, classical upper bounds imply quantum upper bounds. In this thesis we address the question how many quantum examples are *necessary* to learn a concept class $\mathcal{C}$ of VC dimension d. In Chapter 7 we show that the number of quantum examples necessary to learn $\mathcal{C}$ is also given by the VC dimension of $\mathcal{C}$ (improving upon the lower bound of Atıcı and Servedio [AS05]). Combining with the classical upper bound of Hanneke [Han16], our result shows that quantum examples are *not more powerful* than classical examples in the PAC model of learning.

What about more realistic learning models? In many learning situations the examples could possibly be noisy in some way or maybe there is no underlying target concept at all. The *agnostic* model of learning, introduced by Haussler [Hau92] and Kearns et al. [KSS94] takes this into account. It is a well-known result that the sample complexity of agnostic learning a concept class $\mathcal{C}$ is also characterized by the VC dimension of $\mathcal{C}$ (albeit, with a worse dependence on ε). In Chapter 7, we introduce the model of quantum agnostic learning, which wasn't defined prior to our work. We also show that in agnostic learning, quantum examples are *not more powerful* than classical examples.

Part One

Quantum algorithms

Chapter 2

Preliminaries and query complexity

This chapter is divided into two parts. In the first part, we introduce some basic mathematical objects that are used often in this thesis. We then give a brief introduction to quantum information theory. In the second part, we introduce the model of query complexity. We define the decision tree model and the quantum query model. For an excellent survey of different query models and relations between these models, see [BW02] (additionally, the last two years have seen a few breakthroughs [GPW15, ABB+16, ABK16] in the field of classical and quantum query complexity, for a more up-to-date relationship between different query models see Table 2 in [ABK16]). The first new contribution in this thesis (in Chapter 3) will build upon the quantum search algorithm presented in Section 2.5.1 and our second new contribution (in Chapter 4) will give a refinement of the well-known polynomial method presented in Section 2.4.1.

Contents

2.1 Mathematical objects of interest

Set notation. We shall use $\mathbb{R}, \mathbb{C}, \mathbb{N}$ to denote the set of real numbers, complex numbers and natural numbers, respectively. For integer $n > 0$, we denote $[n] = \{1, \ldots, n\}$. Also, the power set of $[n]$ is $2^{[n]} = \{S : S \subseteq [n]\}$.

Bit strings. For $x, y \in \{0,1\}^d$, the bit-wise sum $(x \oplus y) \in \{0,1\}^d$ is the string given by $(x \oplus y)_i = x_i \oplus y_i$. The *Hamming distance* $d(x, y)$ is the number of indices on which x and y differ, $|x \oplus y|$ is the Hamming weight of the string $x \oplus y$ (which equals $d(x, y)$). For $x \in \{0,1\}^d$, denote $\mathrm{supp}(x) = \{i \in [d] : x_i \neq 0\}$. For two bit strings $x, y \in \{0,1\}^d$, we denote the $(2d)$-bit string formed by the concatenation of x and y as (x, y). For $x \in \{0,1\}^d$, let $\bar{x} \in \{0,1\}^d$ be the string given by $(\bar{x})_i = 1 - x_i$.

Distributions. We denote random variables in bold, such as $\mathbf{A}$, $\mathbf{B}$. For a distribution $D : \{0,1\}^n \to [0,1]$, let $\mathrm{supp}(D) = \{x \in \{0,1\}^n : D(x) \neq 0\}$. By $x \sim D$, we mean x is sampled according to the distribution D, i.e., $\Pr[\mathbf{X} = x] = D(x)$.

Polynomials. A monomial is the product of a real coefficient and the product of formal variables, each raised to some power. The degree of a monomial is the sum of the powers of the formal variables. A polynomial is simply a sum of monomials and the degree of a polynomial is the largest degree of its monomials. The simplest class of polynomials are *univariate polynomials* which have only one formal variable. In general, a degree-k univariate polynomial $p : \mathbb{R} \to \mathbb{R}$ is an expression of the form $p(x) = \sum_{i=0}^{k} \alpha_i x^i$ where $\alpha_i \in \mathbb{R}$. More generally, an *n-variate polynomial* (sometimes referred to as *multivariate polynomials* when the number of formal variables is clear) is a polynomial consisting of n formal variables. A degree-k multivariate polynomial $q : \mathbb{R}^n \to \mathbb{R}$ is an expression of the form

$$q(x_1, \ldots, x_n) = \sum_{\substack{i_1, \ldots, i_n \in \{0,1,\ldots,k\}: \\ i_1 + \cdots + i_n \leq k}} \alpha_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n}, \tag{2.1}$$

where $\alpha_{i_1,\ldots,i_n} \in \mathbb{R}$. A special class of polynomials are *multilinear polynomials*, obtained by restricting the sum in Eq. (2.1) to those $i_1, \ldots, i_n$ satisfying $i_j \in \{0,1\}$

for all $j \in [n]$ and $\sum_{j=1}^{n} i_j \leq k$. Alternatively, an n-variate degree-k multilinear polynomial $r : \mathbb{R}^n \to \mathbb{R}$ can be written as

$$r(x_1, \ldots, x_n) = \sum_{\substack{S \subseteq [n]: \\ |S| \leq k}} \alpha_S \prod_{i \in S} x_i,$$

where $\alpha_S \in \mathbb{R}$.

Boolean functions and Fourier analysis. Boolean functions $f : \{0,1\}^n \to \{0,1\}$ are basic objects in theoretical computer science and we use them often in this thesis. For $f : \{0,1\}^n \to \{0,1\}$, we denote

$$f^{-1}(b) = \{x \in \{0,1\}^n : f(x) = b\} \qquad \text{for } b \in \{0,1\}.$$

For Boolean function f, let $\mathrm{supp}(f) = f^{-1}(1)$.

We introduce the basics of Fourier analysis on the Boolean cube $\{0,1\}^n$ here, referring to [O'D14, Wol08] for more. Define the inner product between functions $f, g : \{0,1\}^n \to \mathbb{R}$ as

$$\langle f, g \rangle = \mathop{\mathbb{E}}_{x}[f(x) \cdot g(x)]$$

where the expectation is taken over the uniform distribution on $\{0,1\}^n$. For $S \subseteq [n]$ (equivalently $S \in \{0,1\}^n$),[1] let $\chi_S(x) := (-1)^{S \cdot x}$ denote the parity of the variables (of x) indexed by the set S. It is easy to see that the set of functions $\{\chi_S\}_{S \subseteq [n]}$ form an orthonormal basis for the space of real-valued functions over the Boolean cube. Hence every f can be decomposed as

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S) \chi_S(x) \qquad \text{for all } x \in \{0,1\}^n,$$

where $\widehat{f}(S) = \langle f, \chi_S \rangle = \mathbb{E}_x[f(x) \cdot \chi_S(x)]$ is called a *Fourier coefficient* of f. For a Boolean function $f : \{0,1\}^m \to \{0,1\}$ and $M \in \mathbb{F}_2^{m \times k}$ we define $f \circ M : \{0,1\}^k \to \{0,1\}$ as $(f \circ M)(x) := f(Mx)$ for all $x \in \{0,1\}^k$ (where the matrix-vector product is over $\mathbb{F}_2$).

Vector spaces and matrices. For an n-dimensional vector space, the standard basis of $\mathbb{C}^n$ is denoted by $\{e_i \in \{0,1\}^n : i \in [n]\}$, where e_i is the vector with a 1 in the i-th coordinate and 0's elsewhere. For a matrix $M \in \mathbb{R}^{n \times m}$, let $M^* \in \mathbb{R}^{m \times n}$ be the conjugate transpose of M, i.e., $(M^*)_{ij} = \overline{M_{ij}}$. Let E_{ij} be the elementary matrix defined as $E_{ij} = e_i e_j^*$. A hermitian matrix M is said to be positive semidefinite (psd) if the associated polynomial $x^\mathsf{T} A x$ is non-negative for every vector $x \in \mathbb{R}^n$. Alternatively, we say A is psd if all the eigenvalues of M are non-negative. If M is a psd matrix, we define $\sqrt{M}$ as the unique psd matrix

[1] We will often use this natural bijection between every $S \in \{0,1\}^n$ and $\mathrm{supp}(S) \in 2^{[n]}$.

that satisfies $\sqrt{M} \cdot \sqrt{M} = M$, and $\sqrt{M}(i,j)$ as the (i,j)-th entry of $\sqrt{M}$. For a matrix $A \in \mathbb{R}^{m\times n}$, we denote the singular values of A by $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_{\min\{m,n\}}(A) \geq 0$. Given a set of d-dimensional vectors $U = \{u_1, \ldots, u_n\} \in \mathbb{R}^d$, the Gram matrix V corresponding to the set U is the $n \times n$ psd matrix defined as $V(i,j) = u_i^* u_j$ for $i, j \in [n]$.

For $x \in \mathbb{C}^n$, denote by $\mathrm{Diag}(x)$ the $n \times n$ diagonal matrix whose diagonal forms x. Given a matrix $X \in \mathbb{C}^{n\times n}$ let $\mathrm{diag}(X) \in \mathbb{C}^n$ denote its diagonal vector. For $x \in \{0,1\}^n$, denote $(-1)^x = ((-1)^{x_1}, \ldots, (-1)^{x_n})$. Denote by $\mathbf{1}_d$ the $d \times d$ identity matrix.

Unitary matrices are common in quantum information theory. We say a complex $n \times n$ matrix is a *unitary* if $UU^* = U^*U = \mathbf{1}_n$, where U^* is the conjugate transpose of the matrix U.

The O, o, Ω, Θ notation. This is standard notation in complexity theory. For $f, g : \mathbb{N} \to \mathbb{R}$, we use $f(n) = O(g(n))$ if there exists a constant $c > 0$ and integer N such that $f(n) \leq cg(n)$ for every $n \geq N$. Similarly we write $f(n) = \widetilde{O}(g(n))$ to mean that there exists a constant $k > 0$ such that $f(n) = O(g(n)\log^k(g(n)))$. We write $f(n) = o(g(n))$ to mean $\lim_{n\to\infty} f(n)/g(n) = 0$. We use $f(n) = \Omega(g(n))$ if there exists a constant $c > 0$ and integer N such that $f(n) \geq cg(n)$ for every $n \geq N$. Similarly we write $f(n) = \widetilde{\Omega}(g(n))$ to mean that there exists a constant $k > 0$ such that $f(n) = \Omega(g(n)/\log^k(g(n)))$. Finally, $f(n) = \Theta(g(n))$, if $f(n) = O(g(n))$ and $g(n) = O(f(n))$ and similarly $f(n) = \widetilde{\Theta}(g(n))$, if $f(n) = \widetilde{O}(g(n))$ and $g(n) = \widetilde{O}(f(n))$.

Miscellaneous. We write log for logarithm to base 2, and ln for logarithm to base e. Let $1_{[A]}$ be the indicator for $A \subseteq \{0,1\}^n$, i.e., $1_{[A]}(x)$ is 1 if $x \in A$ and 0 otherwise. Let $\delta_{x,y} = 1_{[x=y]}$. For integer $n > 0$, let S_n be the set of $n!$ permutations on $[n]$.

2.2 Quantum information

In this section we give a general introduction to quantum computation. For more on quantum information we refer to standard textbooks [NC00, KLM06, Wat11] and lectures notes [Wol13, Chi11].

2.2.1 Quantum states

In this section we define qubits, pure quantum states and density operators.

Qubits. Like classical bits are the building blocks of classical computers, quantum bits, common referred to as *qubits*, are the basic building blocks of quantum

computers. Classically, a bit can take the value 0 or 1. The quantum analogue of the bits 0 and 1 are the qubits $|0\rangle \in \mathbb{C}^2$ and $|1\rangle \in \mathbb{C}^2$ respectively,[2] defined as follows

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Observe that $\{|0\rangle, |1\rangle\}$ are the standard basis vectors for $\mathbb{C}^2$, the space in which one-qubit states "live".

Superposition. What makes a qubit different from a classical bit? Quantum mechanics allows a qubit to be in a *superposition*, i.e., a quantum state can informally be *both* in $|0\rangle$ and $|1\rangle$, each associated with an amplitude. In other words a qubit can be in a state of the form

$$|\phi\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \alpha_0|0\rangle + \alpha_1|1\rangle,$$

where $\alpha_0, \alpha_1 \in \mathbb{C}$ satisfy $|\alpha_0|^2 + |\alpha_1|^2 = 1$. The coefficients α_0 and α_1 are referred to as the *amplitudes* of $|0\rangle, |1\rangle$ respectively. The complex conjugate of $|\phi\rangle$ is given by the row vector

$$\langle\phi| = (\alpha_0^* \ \alpha_1^*) = \alpha_0^*\langle 0| + \alpha_1^*\langle 1|.$$

General states. So far we just saw examples of single-qubit systems. Multi-qubit basis states are obtained by taking tensor products of single-qubit basis states; for example, the bases for two-qubit systems are

$$|0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \ |0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \ |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \ |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

where $|0\rangle \otimes |1\rangle$ is the basis state of a 2-qubit system where the first qubit is in state $|0\rangle$ and the second qubit is in state $|1\rangle$. We can extend this definition to arbitrary-dimensional qubit states. For $b \in \{0,1\}^k$, we often shorthand k-qubit state $|b_1\rangle \otimes \cdots \otimes |b_k\rangle$ as $|b_1 \cdots b_k\rangle$ or $|b_1, \ldots, b_k\rangle$. A k-qubit *pure* state $|\phi\rangle$ can be written as $|\phi\rangle = \sum_{i \in \{0,1\}^k} \alpha_i |i\rangle$ where the α_i's are complex numbers (called amplitudes) that satisfy $\sum_{i \in \{0,1\}^k} |\alpha_i|^2 = 1$. We can also view $|\phi\rangle$ as a 2^k-dimensional column vector.

Suppose the k-qubit quantum states $|\phi\rangle$ and $|\psi\rangle$ are represented by unit vectors $u, v \in \mathbb{C}^{2^k}$. The inner product $\langle\phi|\psi\rangle$ is a complex number given by u^*v and the outer product $|\phi\rangle\langle\psi|$ is a complex $2^k \times 2^k$ matrix given by uv^*.

An r-dimensional *quantum state* ρ (also called a *density matrix*) is an $r \times r$ positive semi-definite matrix ρ with trace 1; this can also be written (often non-uniquely) as $\rho = \sum_i p_i |\phi_i\rangle\langle\phi_i|$ and hence can be viewed as a probability distribution over pure states $\{|\phi_i\rangle\}_{i \in [m]}$. We say that ρ is a pure state if ρ satisfies $\text{rank}(\rho) = 1$, i.e., $\rho = |\psi\rangle\langle\psi|$ for some quantum state $|\psi\rangle$.

[2] The $|\cdot\rangle$ is referred to as the "ket" notation and $\langle\cdot|$ is referred as the "bra" notation.

2.2.2 Quantum operations

Given a k-qubit quantum state ρ, what can we do with it? Quantum mechanics allows us to either *evolve* the state to another quantum state σ unitarily or we can *measure* the quantum state.

Quantum gates. Suppose ρ is k-qubit quantum state, then quantum mechanics allows us to apply an arbitrary *unitary transformation* U to ρ. The action of this transformation on ρ is given by $\rho \to U\rho U^*$. If ρ is a pure state, i.e., $\rho = |\psi\rangle\langle\psi|$, then unitary transformation acts by left-multiplication on $|\psi\rangle$, yielding $U|\psi\rangle$. Since every unitary U has an inverse U^*, it follows that every unitary transformation on quantum states is reversible: one could simply apply $U^{-1} = U^*$ and recover ρ (from $U\rho U^*$) without losing any information about ρ.

Quantum gates are usually unitaries, acting on at most three qubits. An important class of single-qubit gates are the *Pauli operators* $\{\mathbf{1}_2, X, Y, Z\}$, whose action is given by:

$$\mathbf{1}_2 : |b\rangle \mapsto |b\rangle, \quad X : |b\rangle \mapsto |b\oplus 1\rangle, \quad Y : |b\rangle \mapsto (-1)^b i|b\oplus 1\rangle, \quad Z : |b\rangle \mapsto (-1)^b|b\rangle$$

for $b \in \{0,1\}$. The corresponding matrix representation of these Pauli matrices is given by

$$\mathbf{1}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

In the circuit model these gates are represented as follows

Figure 2.1: Pauli gates $\{\mathbf{1}_2, X, Y, Z\}$ in the circuit model

The single-qubit Hadamard transform corresponds to the unitary map

$$H : |a\rangle \mapsto \frac{|0\rangle + (-1)^a|1\rangle}{\sqrt{2}} \quad \text{for } a \in \{0,1\}.$$

We often shorthand the notation $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ as $|+\rangle$ and $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ as $|-\rangle$. The matrix representation of the Hadamard transform is given by

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

The two-qubit controlled-not gate (referred to as the CNOT gate) corresponds to the following map: for $b_1, b_2 \in \{0,1\}$, on input $|b_1, b_2\rangle$, the CNOT gate flips b_2

if b_1 is 1 and does nothing to b_2 if b_1 is 0,

$$\text{CNOT} : |0, b_2\rangle \mapsto |0, b_2\rangle$$
$$\text{CNOT} : |1, b_2\rangle \mapsto |1, b_2 \oplus 1\rangle.$$

The unitary corresponding to the CNOT gate is given by

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

The Hadamard and CNOT gate in the circuit model are represented as follows

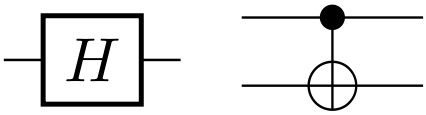


Figure 2.2: Hadamard and CNOT gate in the circuit model

The CNOT gate is sometimes referred to as the controlled-X gate, since on input $|b_1, b_2\rangle$, the CNOT gate applies X^{b_1} on the second qubit (where X^{b_1} is the matrix X raised to the power b_1). For an n-qubit unitary U, one can generalize the controlled-X gate definition, and define a controlled-U gate acting on $n+1$ qubits as follows: for every $b_1, \ldots, b_{n+1} \in \{0, 1\}$,

$$\text{controlled-}U : |b_1\rangle|b_2, \ldots, b_{n+1}\rangle \mapsto |b_1\rangle U^{b_1}|b_2, \ldots, b_{n+1}\rangle.$$

The controlled-U gate in the circuit model is represented as follows

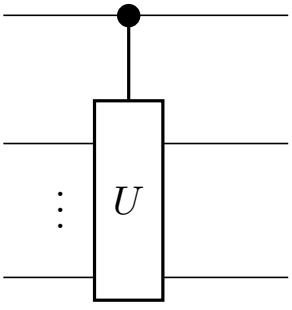


Figure 2.3: For an n-qubit unitary U, the controlled-U gate in the circuit model

The three-qubit Toffoli gate is also called the controlled-controlled NOT gate (referred to as the CCNOT gate). The Toffoli gate flips the last input bit if and only if the first two input bits are 1:

$$\text{CCNOT} : |b_1, b_2, b_3\rangle \mapsto |b_1, b_2, b_3 \oplus (b_1 \cdot b_2)\rangle \quad \text{for every } b_1, b_2, b_3 \in \{0, 1\}.$$

Finally, most quantum algorithms in this thesis will begin by creating the uniform superposition state using the n-qubit Hadamard transform, denoted as $H^{\otimes n}$, whose action on $|0^n\rangle$ is given by

$$H^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{2^n}} \sum_{a\in\{0,1\}^n} |a\rangle.$$

Measurements. Although quantum mechanics allows quantum states to be in a superposition, an unfortunate aspect of quantum mechanics is that, given a quantum system containing an *unknown* k-qubit state $|\phi\rangle$, it is not possible to retrieve the 2^k-dimensional vector corresponding to $|\phi\rangle$. In order to obtain any information from a quantum system, we need to *measure* the system. Suppose we are given a quantum system in the state $|\phi\rangle = \sum_{i\in\{0,1\}^k} \alpha_i|i\rangle$ and we would like to obtain classical information from $|\phi\rangle$. One naïve possibility is to simply measure the k qubits. Then, quantum mechanics predicts that we will see an outcome $j \in \{0,1\}^k$ with probability $|\alpha_j|^2$. Since $|\phi\rangle$ was a unit vector in $\mathbb{C}^{2^k}$, measuring $|\phi\rangle$ in the computational basis can be viewed as simply sampling from the distribution given by the squared-amplitude distribution $\{|\alpha_i|^2\}_{i\in[2^k]}$. Suppose we measure $|\phi\rangle$ and saw the outcome j, then $|\phi\rangle$ *collapses* to $|j\rangle$, which cannot be reused to obtain any other information about $|\phi\rangle$.

The measurement operation in the circuit model is represented as follows

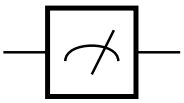


Figure 2.4: Measurement operation (in the computational basis) in the circuit model

Of course, there is more to quantum mechanics than just computational basis measurement. In general, suppose we have a quantum state ρ, then to obtain classical information from ρ, one can apply an m-outcome quantum measurement, also called a *POVM* (positive-operator-valued measure). This POVM is described by a set of positive semi-definite matrices $\{M_i\}_{i\in[m]}$ that satisfy $\sum_i M_i = \mathbf{1}$. When measuring ρ using this POVM, the probability of outcome j is given by $\mathrm{Tr}(M_j\rho)$ and the resulting quantum state after the measurement is $M_j\rho M_j^*/\mathrm{Tr}(M_j\rho)$.

2.3 Query models

In this section we begin by discussing the classical decision tree model and its quantum generalization, the quantum query model. The central object here is a Boolean function f. Let $\mathcal{D} \subseteq \{0,1\}^n$ and suppose $f : \mathcal{D} \to \{0,1\}$ is a Boolean

function. We say f is a *total Boolean function* if $\mathcal{D} = \{0,1\}^n$, i.e., $f(x)$ is defined on every point on the Boolean cube $\{0,1\}^n$. Suppose $\mathcal{D} \subset \{0,1\}^n$, then we say that f is a *partial Boolean function*. The need for this distinction between partial and total Boolean functions will become clear when we discuss relationships between quantum and classical query complexity. In both query models, the goal is to compute $f(x)$ for an unknown x. The only difference between these models is the *query access* to the unknown x. As the name suggests, in the decision tree model, we are allowed to make classical queries and in the quantum query model, we can make queries in a quantum superposition.

2.3.1 Decision tree complexity

We now define the deterministic query complexity of f. Promised that $x \in \mathcal{D}$, the goal is to learn $f(x)$, when only given access to x through an *oracle* that encodes x. An application of the oracle is usually referred to as a *query*, which consists of an index $i \in [n]$ and the response of the oracle $x_i \in \{0,1\}$.[3] A *deterministic decision tree* is a binary tree $\mathcal{A}$. Each node in $\mathcal{A}$ is labeled with some $i \in [n]$ and has two outgoing edges, labeled 0 and 1 (depending on the value of x_i). The leaves of $\mathcal{A}$ are labeled with an output bit $\{0,1\}$. Given an input $x = x_1 \cdots x_n$, the tree proceeds at the ith node by evaluating the input bit x_i and continuing in the subtree corresponding to the value of $x_i \in \{0,1\}$. The output of $\mathcal{A}$ is the value of the leaf that is reached eventually. Apart from making queries, $\mathcal{A}$ is also allowed to perform arbitrary reversible operations in between queries depending on the response to the previous queries.

We say an decision tree $\mathcal{A}$ *exactly computes* f if the output of the tree $\mathcal{A}(x)$ equals $f(x)$ on every input $x \in \mathcal{D}$. Note that we are not concerned with the output of the algorithm for the xs not in $\mathcal{D}$. The cost of $\mathcal{A}$ on input x, denoted $C(\mathcal{A}, x)$, is the number of queries that $\mathcal{A}$ makes to the oracle encoding x. Clearly there are different decision trees that compute f. The *deterministic query complexity* of f, denoted $D(f)$, is the "worst-case" cost of the "best" decision tree that computes f, i.e.,

$$D(f) = \min_{\mathcal{A}} \max_{x} C(\mathcal{A}, x),$$

where the first minimum is over all decision trees $\mathcal{A}$ that exactly compute f and the maximum is over $x \in \mathcal{D}$.

Clearly $D(f) \leq n$ for every f since the decision tree $\mathcal{A}$ could simply query every bit of $x \in \{0,1\}^n$ and compute $f(x)$. Also $D(f) \geq 0$ for every f. Both these inequalities are tight, the latter being true for the constant 1 function. In order to exhibit a function satisfying $D(f) = n$, let $f : \{0,1\}^n \to \{0,1\}$ be the function defined as follows: $f(x)$ equals 1 if and only if there exists an $i \in [n]$ such that

[3] A natural way to model a classical query to the oracle is by the reversible map that maps an input (i, b) to $(i, b \oplus x_i)$ for $i \in [n]$ and $b \in \{0,1\}$.

$x_i = 1$. This function is also referred to as the OR_n function on n bits. We claim that $D(\mathrm{OR}_n) = n$. Let $\mathcal{A}$ be the best decision tree minimizing the outer min in $D(f)$. Suppose $\mathcal{A}$ makes $n-1$ queries to the oracle and observes 0 always, then $\mathcal{A}$ knows x almost entirely, except for one bit of x (say, x_n is unknown). Observe that the value of x_n could be 1 in which case $\mathrm{OR}_n(0^{n-1}, 1)$ evaluates to 1, or x_n could be 0 in which case $\mathrm{OR}_n(0^{n-1}, 0)$ evaluates to 0. This forces $\mathcal{A}$ to make n queries on the worst input in order to compute OR_n with certainty.

Randomized query model. The randomized query model adds the power of randomization to the decision tree model and tries to understand if randomness can reduce query complexity. A *randomized decision tree* $\mathcal{A}_\mu$ is defined as a probability distribution μ over deterministic decision trees. On input x, a randomized decision tree first samples a deterministic decision tree $\mathcal{A}$ according to μ and then outputs $\mathcal{A}(x)$. The expected cost of $\mathcal{A}_\mu$ is then defined as

$$C(\mathcal{A}_\mu) = \mathop{\mathbb{E}}_{\mathcal{A}\sim\mu} \max_{x\in\{0,1\}^n} C(\mathcal{A}, x).$$

A *two-sided* randomized decision tree is said to compute $f : \mathcal{D} \to \{0,1\}$ with error $\varepsilon \geq 0$, if the output of $\mathcal{A}_\mu$, on input $x \in \mathcal{D}$, is equal to $f(x)$ with probability at least $1 - \varepsilon$ (where the probability is taken over the randomness of the algorithm). Finally, the *randomized query complexity*, denoted $R_\varepsilon(f)$, is the minimum expected cost of a two-sided randomized decision tree that computes f with error ε. For notational convenience, we write $R(f)$ as a shorthand for $R_{1/3}(f)$.[4]

Clearly $R(f) \leq D(f)$. In fact, it is not hard to see that there could be a saving in this model. For example, $R(\mathrm{OR}_n) \leq 2n/3$. Indeed, consider a randomized algorithm that samples $2n/3$ indices $i \in [n]$ uniformly at random and queries the x_is corresponding to the is it has seen. Suppose the algorithm finds a j such that $x_j = 1$, then it outputs 1, if not it outputs 0. It is not hard to see that this algorithm computes OR_n with probability at least $2/3$, so $R(\mathrm{OR}_n) \leq 2n/3$.

2.3.2 Quantum query complexity

The quantum query model was formally defined by Beals et al. [BBC+01]. In this model, we are given black-box access to a unitary operator, often called an oracle O_x, whose description depends in a simple way on some binary input string $x \in \{0,1\}^n$. An application of the oracle on a quantum register is referred to as a quantum *query* to x. In the standard form of the model, a query acts on a pair of registers (A, Q), where A is a one-qubit auxiliary register and Q is an n-dimensional query register. A quantum query to the oracle is a coherent version

[4]Note that $1/3$ is an arbitrary constant often used when discussing query complexity. One could simply reduce the error probability to ε by repeating the algorithm $O(\log(1/\varepsilon))$ times and taking the majority of the output.

the classical query and corresponds to the unitary transformation given by

$$O_x : |b, i\rangle \to |b \oplus x_i, i\rangle,$$

where $b \in \{0, 1\}$ and $i \in [n]$. These oracles are also commonly called *bit oracles*.

Apart from the queries O_x, a quantum query algorithm consists of a fixed sequence of unitary operations acting on registers $(\mathsf{A}, \mathsf{Q}, \mathsf{W})$, where W is an additional *workspace* register. A t-query quantum algorithm begins by initializing the joint register $(\mathsf{A}, \mathsf{Q}, \mathsf{W})$ in the all-zero state and continues by interleaving a sequence of unitaries $U_0, \ldots, U_t$ on $(\mathsf{A}, \mathsf{Q}, \mathsf{W})$ with oracles O_x acting on (A, Q). The algorithm concludes by measuring the first register A (as in the figure below) and returns the measurement outcome.[5]

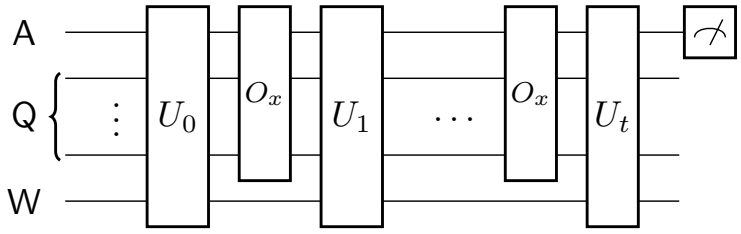


Figure 2.5: A t-query quantum algorithm that begins in the all-zero state and outputs the measurement outcome of register A

For a Boolean function $f : \{0, 1\}^n \to \{0, 1\}$, the algorithm is said to compute f with error $\varepsilon \geq 0$ if for every x, the measurement outcome of register A equals $f(x)$ with probability at least $1 - \varepsilon$. The *bounded-error query complexity* of f, denoted $Q_\varepsilon(f)$, is the smallest t for which such an algorithm exists. Note that in the query model, we are not concerned with the amount of time (i.e., the number of gates) it takes to implement the interlacing unitaries, which could be much bigger than the query complexity itself. From here onwards, unless explicitly mentioned, we write $Q(f)$ as a shorthand for $Q_{1/3}(f)$. Clearly $Q(f) \leq R(f)$. In fact, we will see in the next section that for $f = \mathrm{OR}_n$, $Q(f)$ is quadratically smaller than $R(f)$.

For convenience, we will often work with a slightly less standard oracle sometimes referred to as a *phase oracle*, denoted by $O_{x,\pm}$. The action of $O_{x,\pm}$ on the basis states can be described by

$$O_{x,\pm} : |b, i\rangle \to (-1)^{b \cdot x_i} |b, i\rangle, \tag{2.2}$$

where $b \cdot x_i$ is the bitwise AND between b and x_i. Using the standard "phase kick-back trick", the phase oracle can be obtained from the standard oracle O_x, preceded and followed by a Hadamard on A. Indeed, given access to the bit

[5] Without loss of generality, we can assume that all intermediate measurements can be deferred to the end of the circuit.

oracle O_x, we can make a phase query as follows: start with $|b,i\rangle$ and apply the Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ to the first qubit to obtain $(|0\rangle|i\rangle + (-1)^b|1\rangle|i\rangle)/\sqrt{2}$. Apply O_x to obtain

$$\frac{|x_i\rangle|i\rangle + (-1)^b|1 \oplus x_i\rangle|i\rangle}{\sqrt{2}}.$$

Finally, apply the Hadamard gate to the first qubit and observe that the resulting state is $(-1)^{b \cdot x_i}|b,i\rangle$.

Since the Hadamards involved in the intercoversion between O_x and $O_{x,\pm}$ can be undone by the unitaries surrounding the queries in a quantum query algorithm, using the phase oracle does not reduce generality.

2.3.3 Separations between quantum and classical query complexity

One of the main attractions of the query model is its simplicity. In order to analyze the complexity, we need not worry about the time taken by the algorithm, the amount of space used and the gates/operations involved in the algorithm. In contrast to other models like circuit complexity, the simplicity in the query model allows one to prove tight (unconditional) lower bounds and show provable separations between different query models.

One area of study which has received a lot of attention is the relationship between the quantum and classical query models: can the quantum query model give exponential savings compared to the deterministic query model for certain Boolean functions or is there at most a polynomial advantange in the quantum query model for a class of Boolean functions? Two famous results in this direction were the period-finding algorithm (which was a crucial subroutine in Shor's factoring algorithm [Sho97]) that finds the period of a periodic function exponentially faster than every classical algorithm, and Grover's search algorithm [Gro96] which finds a marked item in an unstructured database using quadratically fewer quantum queries than a classical search algorithm. Although the speed-up for the search problem is only quadratic, the search algorithm has found application in numerous examples. We discuss this further in Section 2.5.1.

We briefly comment on the state-of-the-art separations between $R(f), D(f)$ and $Q(f)$. For partial Boolean functions, we know of exponential separations between these measures. The first exponential separation between $D(f)$ and $Q(f)$ was given by Deutsch and Josza [DJ92]. They exhibited a partial function on N bits which could be solved using 1 quantum query and showed that every classical deterministic algorithm would need to make $\Omega(N)$ queries. However, a randomized algorithm could solve the Deutsch-Jozsa problem efficiently if allowed a small error probability, so this function didn't separate $R(f)$ and $Q(f)$. Subsequently,

Simon [Sim97] defined a partial function which gave an exponential separation between $R(f)$ and $Q(f)$. Larger separations between $R(f)$ and $Q(f)$ were obtained (and conjectured) in more recent results of Aaronson and Ambainis [AA15] using the so-called Forrelation problem and the k-fold variant of the Forrelation problem.

The situation is completely different for *total* Boolean functions. Beals et al. [BBC+01] ruled out the possibility of an exponential speed-up between $Q(f)$ and $D(f)$ and showed that for total functions, there is at most a polynomial separation between $D(f), R(f), Q(f)$. It was shown that $D(f) \leq Q(f)^6$ [BBC+01] and $D(f) \leq R(f)^3$ [Nis91] for all total Boolean functions. The question that remains open is: are these inequalities tight, i.e., do there exist total Boolean functions f, g for which $D(f) \geq Q(f)^6$ and $D(g) \geq R(g)^3$? The largest separation between $D(f)$ and $Q(f)$ was a quadratic separation for $f = \mathrm{OR}_n$, for which $D(f) \geq R(f) \geq Q(f)^2$ (since Grover's search algorithm [Gro96] shows $Q(\mathrm{OR}_n) \leq O(\sqrt{n})$ and $R(\mathrm{OR}_n) = \Omega(n)$). Larger separations were not known until recently. In 2015, Ambainis et al. [ABB+16] in a breakthrough result, showed among many things, explicit total Boolean functions f and g that satisfy $D(f) \geq Q(f)^4$ and $D(g) \geq R(g)^2$. The Boolean function they used to separate these measures was inspired by the so-called "pointer functions" introduced in the work of Göös, Pittasi and Watson [GPW15].

2.4 Lower bound methods for quantum query complexity

Several methods have been proposed over the years to give upper bounds on quantum query complexity. Almost always, one gives an upper bound on query complexity by explicitly constructing an algorithm that solves the problem and analyzing the complexity of the algorithm. In this direction there are a few general methods used often to construct quantum algorithms, such as quantum walks [Amb07, MNRS11], span programs [Rei09], learning graphs [Bel12], so-called bomb query complexity [LL16]. However, in addition to understanding the advantage provided by quantum query algorithms, it is also important to understand their *limitations*, which requires proving *lower bounds* on query complexity. In order to prove a lower bound, we need to show that *every* algorithm that solves a problem needs to make at least a certain number of queries. Proving such a statement seems very hard since we need to argue about *all* possible algorithms, in fact its not even clear how one would prove such a statement.

There are two well-known methods known to give lower bounds for quantum query complexity: polynomial method introduced by Beals et al. [BBC+01] and the adversary method introduced by Ambainis [Amb00]. The latter was eventually generalized to the so-called negative-weight adversary method [HLŠ07] and was shown to *characterize* quantum query complexity [HLŠ07, Rei09, Rei11,

LMR+11]. In the next two sections we give a brief overview of these lower bound techniques.

2.4.1 Polynomial method

Beals et al. [BBC+01] made the following simple, yet beautiful connection between multilinear polynomials on the Boolean cube and quantum query algorithms, which gave rise to the polynomial method.

2.4.1. Lemma. *Suppose $\mathcal{A}$ is a t-query quantum algorithm given oracle access to an input $x \in \{0,1\}^n$, then the acceptance probability of $\mathcal{A}$ is a degree-$(2t)$ real multilinear polynomial in $x_1, \ldots, x_n$.*

Proof. Beals et al. [BBC+01] in fact observed something stronger, the amplitude of *every* basis state in $\mathcal{A}$ after making t queries is a degree-t complex multilinear polynomial. More precisely, they showed the final state of $\mathcal{A}$ (in terms of Figure 2.5) before the measurement of register A can be written as

$$\sum_{b\in\{0,1\},i\in[n],w\in\{0,1\}^m} \alpha_{b,i,w}(x)|b,i,w\rangle,$$

where $\alpha_{i,b,w}(x)$ is a degree-t complex multilinear polynomial in $x_1, \ldots, x_n$ and W is assumed to act on m workspace qubits. We now prove this by induction on t.

Base case. Clearly when $t = 0$, $\mathcal{A}$ hasn't made any queries to $x \in \{0,1\}^n$, so the amplitude of the basis states depend only on U_0 and are *independent* of x. In particular, the amplitudes can be viewed as degree-0 polynomials.

Induction hypothesis. Suppose after $t-1$ queries, the state of the quantum algorithm is given by

$$\sum_{b\in\{0,1\},i\in[n],w\in\{0,1\}^m} \beta_{b,i,w}(x)|b,i,w\rangle,$$

where $\beta_{b,i,w}(x)$ are complex multilinear polynomials of degree at most $t-1$.

Induction step. Suppose $\mathcal{A}$ makes one more query. The action of $(O_{x,\pm} \otimes \mathbf{1}_{\mathsf{W}})$ on a basis state $|b,i,w\rangle$ with amplitude $\beta_{b,i,w}(x)$ can be written as

$$\begin{aligned}(O_{x,\pm} \otimes \mathbf{1}_{\mathsf{W}}) \cdot \beta_{b,i,w}(x)|b,i,w\rangle &= \beta_{b,i,w}(x)(-1)^{b\cdot x_i}|b,i,w\rangle \\ &= \beta_{b,i,w}(x)(1-2b\cdot x_i)|b,i,w\rangle.\end{aligned}$$

Let $\alpha_{b,i,w}(x) := \beta_{b,i,w}(x)(1-2b\cdot x_i)$ be the new amplitude of the basis state $|b,i,w\rangle$ after t queries. Clearly $\alpha_{b,i,w}(x)$ has degree at most 1 more than the degree

of $\beta_{b,i,w}(x)$, so at most t. The polynomial $\alpha_{b,i,w}(x)$ can be made multilinear because of the relation $x_i^2 = x_i$ for $x_i \in \{0,1\}$. Finally, the unitary U_t only redistributes the amplitudes of the basis states in a linear way and is independent of x, thereby not increasing the degree of the polynomials in the amplitude. This concludes the induction step and shows that the amplitudes of $\mathcal{A}$ are degree-t complex polynomials in $x_1, \ldots, x_n$.

In order to conclude the proof of the lemma, simply observe that the probability $\mathcal{A}$ accepts is given by the probability that the measurement of register A (in Figure 2.5) results in 1, which is given by

$$\Pr[\mathcal{A} \text{ outputs } 1] = \sum_{i \in [n], w \in \{0,1\}^m} |\alpha_{1,i,w}(x)|^2. \tag{2.3}$$

This is clearly a real multilinear polynomial of degree at most $2t$ (since the degree of $\alpha_{b,i,w}(x)$ was at most t). □

In order to put this lemma to use, we need the following definition of approximate degree of a Boolean function f.

2.4.2. Definition (Approximate degree). Let $\mathcal{D} \subseteq \{0,1\}^n$ and $\varepsilon \geq 0$. The *ε-approximate degree* of $f : \mathcal{D} \to \{0,1\}$, denoted $\deg_\varepsilon(f)$, is the smallest positive integer k for which there exists a degree-k multilinear polynomial $p : \mathbb{R}^n \to \mathbb{R}$ such that

1. $|p(x) - f(x)| \leq \varepsilon$ *for every* $x \in \mathcal{D}$
2. $|p(x)| \leq 1$ *for every* $x \in \{0,1\}^n$.

For simplicity, let $q(x) = \Pr[\mathcal{A} \text{ outputs } 1]$ in Eq. (2.3) and let us assume $\deg(q) \leq 2t$. Suppose that $\mathcal{A}$ is an optimal t-query quantum algorithm that computes a Boolean function $f : \mathcal{D} \to \{0,1\}$ with error at most $1/3$. It follows that: for all $x \in \mathcal{D}$, if $f(x) = 0$, then $q(x) \in [0, 1/3]$ and if $f(x) = 1$, then $q(x) \in [2/3, 1]$ (where we used that $q(x) \in [0,1]$). Clearly q satisfies the requirements of Definition 2.4.2 for $\varepsilon = 1/3$, hence $Q(f) = t \geq \deg(q)/2 \geq \deg_{1/3}(f)/2$. In general, the polynomial method gives us the following corollary.

2.4.3. Corollary. *Let $\varepsilon \geq 0$ and $\mathcal{D} \subseteq \{0,1\}^n$. Then for every $f : \mathcal{D} \to \{0,1\}$, we have that $Q_\varepsilon(f) \geq \deg_\varepsilon(f)/2$.*

The polynomial method thus converts the problem of lower bounding quantum query complexity to the problem of proving lower bounds on $\deg_\varepsilon(f)$. Given that lower bounds on approximate degree (in particular for univariate polynomials) have been studied for decades in the field of approximation theory, this corollary allows us to use results from the latter to show lower bounds on quantum query complexity. In fact, in Section 2.5.2, we show that Grover's search algorithm is optimal for computing the OR_n function, using fundamental results from approximation theory.

2.4.2 Adversary method

In this section, we discuss another approach to show lower bounds on quantum query complexity, the *quantum adversary method*. The original adversary method was introduced by Ambainis [Amb00] and is often refered to as the "positive-weight" adversary method. However, it was later shown that this method cannot be used to prove optimal lower bounds for certain Boolean functions [Zha05, ŠS06].[6] Høyer, Lee and Špalek [HLŠ07] extended the work of Ambainis and defined the "negative-weight" adversary method, which can prove strictly better quantum query lower bounds than the bounds obtained by the positive-weight adversary method. Subsequently, a series of works by Reichardt and others [Rei09, Rei11, LMR+11] showed that the negative-weight adversary method in fact *characterizes* quantum query complexity, i.e., the negative-weight adversary method can also be used to give *upper bounds* on quantum query complexity. In this section, we describe the lower bound obtained by the positive-weight adversary method. The negative-weight method is significantly more complicated and we refer the reader to [HLŠ07],[Chi11, Chapter 22].

Let's recall the quantum query model discussed in Section 2.3.2. The final state (before the measurement) of a t-query quantum algorithm computing a Boolean function f can be written as

$$|\psi_x^t\rangle = U_t O_x U_{t-1} \cdots U_1 O_x U_0 |0 \cdots 0\rangle.$$

The basic idea of the adversary method is the following: a good T-query quantum algorithm for f should be able to differentiate between an $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$, i.e., it should be able to distinguish between $|\psi_x^T\rangle$ and $|\psi_y^T\rangle$. If the algorithm has made no queries, then it cannot distinguish between x and y and clearly $|\langle\psi_x^0|\psi_y^0\rangle| = 1$. However, every oracle application gives "some information" about the input string. After T queries, a good algorithm (with error $\leq 1/3$) should be able to distinguish between $|\psi_x^T\rangle$ and $|\psi_y^T\rangle$ with probability at least $2/3$. Using the following well-known fact in quantum information theory, which can be found for instance in [KLM06, Theorem A.9.1], it follows that $|\langle\psi_x^T|\psi_y^T\rangle|$ is significantly smaller than 1.

2.4.4. Fact. Let binary random variable $\mathbf{b} \in \{0,1\}$ be uniformly distributed. Suppose an algorithm is given $|\psi_{\mathbf{b}}\rangle$ (for unknown b) and is required to guess whether $\mathbf{b} = 0$ or $\mathbf{b} = 1$. It will guess correctly with probability at most $\frac{1}{2} + \frac{1}{2}\sqrt{1 - |\langle\psi_0|\psi_1\rangle|^2}$. In particular, if we can distinguish $|\psi_0\rangle$ and $|\psi_1\rangle$ with probability $\geq 1 - \delta$, then $|\langle\psi_0|\psi_1\rangle| \leq 2\sqrt{\delta(1-\delta)}$.

Indeed, by plugging in $\delta = 2/3$, we get $|\langle\psi_x^0|\psi_y^0\rangle| \leq 17/18$. In T queries, the algorithm goes from having no information about x, y (i.e., $|\langle\psi_x^0|\psi_y^0\rangle| = 1$) to

[6] This is often referred to as the certificate barrier, a combinatorial object which we do not define here.

distinguishing x and y with probability $\geq 2/3$ (i.e., $|\langle\psi_x^T|\psi_y^T\rangle| \leq 17/18$). Suppose we can upper bound the amount of information provided by a single oracle application, then we can use this to give a lower bound on T. We make this formal now.

Instead of picking a single $(x, y) \in f^{-1}(0)\times f^{-1}(1)$ as above, Ambainis [Amb02] suggested considering a subset $R \subseteq f^{-1}(0) \times f^{-1}(1)$ of hard-to-distinguish (x, y)-pairs and defining a *progress measure* as follows:

$$\Phi(t) = \sum_{(x,y)\in R} |\langle\psi_x^t|\psi_y^t\rangle|.$$

Let us make a few observations about the progress measure function Φ. First observe that $\Phi(t)$ is not affected by the application of a unitary since unitary transformations preserve inner product (by definition). Hence, it follows that $\Phi(0) = |R|$, since $|\psi_x^0\rangle$ and $|\psi_y^0\rangle$ are independent of x, y respectively.

Suppose $Q(f) = T$. Then, there exists a T-query quantum algorithm that can distinguish $|\psi_x^T\rangle$ and $|\psi_y^T\rangle$ with probability at least $2/3$ for every $(x, y) \in f^{-1}(0) \times f^{-1}(1)$. In particular, using $\delta = 1/3$ in Fact 2.4.4, we get $|\langle\psi_x^T|\psi_y^T\rangle| \leq 17/18$ for every $(x, y) \in f^{-1}(0) \times f^{-1}(1)$. Hence,

$$\Phi(T) = \sum_{(x,y)\in R} |\langle\psi_x^T|\psi_y^T\rangle| \leq 17|R|/18.$$

This implies that $|\Phi(T) - \Phi(0)| \geq |R|/18$. Suppose we could show that the change in the progress measure in *every* step can be upper bounded by Δ, i.e., $|\Phi(t + 1) - \Phi(t)| \leq \Delta$ for every t, then we obtain a lower bound on T. To see this, observe that

$$\begin{aligned}|\Phi(T) - \Phi(0)| &= |\Phi(T) - \Phi(T-1) + \Phi(T-1) + \cdots - \Phi(1) + \Phi(1) - \Phi(0)| \\ &\leq |\Phi(T) - \Phi(T-1)| + \cdots + |\Phi(1) - \Phi(0)| \\ &\leq T\Delta,\end{aligned}$$

where the first inequality follows from triangle inequality. Putting together $|\Phi(T) - \Phi(0)| \geq |R|/18$ and the inequality above, we get $T \geq |R|/(18\Delta)$. Ambainis [Amb00] used this idea and proved the following theorem.

2.4.5. Theorem. *Let $\mathcal{D} \subseteq \{0, 1\}^n$ and $f : \mathcal{D} \to \{0, 1\}$. Suppose R is a relation $R \subseteq f^{-1}(0) \times f^{-1}(1)$ (equivalently, a bipartite graph with vertices labelled by n-bit strings) with the following properties:*

1. *Every left-vertex v is related to at least m right-vertices w (i.e., $|\{w \in f^{-1}(1) : (v, w) \in R\}| \geq m$).*
2. *Every right-vertex w is related to at least m' left-vertices v (i.e., $|\{v \in f^{-1}(0) : (v, w) \in R\}| \geq m'$).*

3. *For every $i \in [N]$, every left-vertex v is related to at most ℓ right-vertices w satisfying $v_i \neq w_i$.*

4. *For every $i \in [N]$, every right-vertex w is related to at most ℓ' left-vertices v satisfying $v_i \neq w_i$.*

Suppose there exists a quantum algorithm that, on input $x \in \mathcal{D}$, outputs $f(x)$ with high probability, then

$$\Delta \leq O\Big(\sqrt{\frac{\ell\ell'}{mm'}}|R|\Big),$$

and therefore, the algorithm makes at least $|R|/(18\Delta) = \Omega(\sqrt{mm'/\ell\ell'})$ queries.

We do not prove the theorem here and refer the interested reader to Ambainis [Amb00]. Although the theorem is fairly simple to state, in order to obtain interesting lower bounds using this method, it is important to cleverly choose the relation R that simultaneously maximizes m, m' and minimizes ℓ, ℓ'. We see one such example in Section 2.5.2.

2.5 Quantum search in a database

One of the main successes of quantum algorithms so far is Grover's algorithm for *database search* [Gro96, BHMT02]. Here a database of size N is modeled as a binary string $x \in \{0,1\}^N$, whose bits are indexed by $i \in \{0, \ldots, N-1\}$. A *solution* is an index i such that $x_i = 1$. The goal of the search problem is to find such a solution given query access to x. A *decision* version of the problem asks if there *exists* a solution. Note that this decision problem is equivalent to computing the function $\mathrm{OR}_N : \{0,1\}^N \to \{0,1\}$ which, on input x, evaluates to 1 if and only if $|x| \geq 1$.

If our database has Hamming weight $|x| = 1$, we say it has a *unique* solution. In this case, it is not hard to see that one needs to make $\Omega(N)$ classical queries in order to find a solution. In the quantum setting, Grover discovered a surprising quantum algorithm that finds a solution with high probability using $O(\sqrt{N})$ database queries and $O(\sqrt{N}\log N)$ other elementary gates. For the special case of a database with a unique solution the number of queries is essentially $\frac{\pi}{4}\sqrt{N}$, and Zalka [Zal99] showed that this number of queries is optimal. There are variations of the search problem wherein the database has at least t solutions, and t is a parameter that could possibly be unknown to the algorithm. In this case, there is a variant of Grover's algorithm that, with high probability, finds *a* solution using $O(\sqrt{N/t})$ queries and another variant that finds *all* the solutions using $O(\sqrt{Nt})$ queries. For a quick summary of the important variations of Grover's algorithm, we refer the reader to [Wol10, Appendix A].

While Grover's search algorithm does not provide an exponential speed-up like Shor's factoring algorithm [Sho97] and might seem not-so impressive, the

search algorithm in various forms and generalizations has been applied as a subroutine in many other quantum algorithms. Often the main source of polynomial speed-ups for these algorithms is due to Grover's search algorithm. See for example [BHT97, BCW98, Amb04, BDH+05, DH96, DHHM06, Dör07, Kot14, LL16, LMP15, Mon17b].

2.5.1 Grover's algorithm

We now formally describe Grover's search algorithm for the following problem:

2.5.1. Definition (*Unstructured search*). Let $n \in \mathbb{N}$ and $N = 2^n$. Given a database modelled as $x \in \{0,1\}^N$, the goal is to find an index $i \in [N]$ such that $x_i = 1$ (we refer to such an i as a solution) and output 'no solutions' if there exists no such i.

In order to solve the problem, we assume that we can access the database by means of phase oracle $O_{x,\pm}$. In fact we will assume something more about these phase oracles. Grover's algorithm consists of a sequence of unitaries $U_0, \ldots, U_t$ (as in Figure 2.5) which do not act on register A. Hence we can set register A to the state $|1\rangle$. From here onwards, we will ignore register A in Fig. 2.5 and let the phase oracle correspond to the transformation $O_{x,\pm} : |i\rangle \to (-1)^{x_i}|i\rangle$.

In order to describe Grover's algorithm we need the *diffusion operator*

$$D = H^{\otimes n}(2|0^n\rangle\langle 0^n| - \mathbf{1}_N)H^{\otimes n} = 2|\psi\rangle\langle\psi| - \mathbf{1}_N,$$

where $|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{i\in\{0,1\}^n}|i\rangle$ is the uniform superposition state. Grover's algorithm can then be described by circuit in Figure 2.6.

Let us now verify the correctness of the algorithm. For simplicity, assume that there is a unique solution at index k, i.e., $x_k = 1$. In order to understand the action of $DO_{x_\pm}$ on $|\psi\rangle$, it is convenient to introduce a "good" state $|G\rangle$ and "bad" state $|B\rangle$,

$$|G\rangle = |k\rangle, \quad |B\rangle = \frac{1}{\sqrt{N-1}}\sum_{i\in\{0,1\}^n\setminus\{k\}}|i\rangle.$$

The state of the algorithm after the first batch of Hadamard gates can then be written as

$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{i\in\{0,1\}^n}|i\rangle = \frac{1}{\sqrt{N}}|G\rangle + \sqrt{\frac{N-1}{N}}|B\rangle \tag{2.4}$$

The whole point of defining $|G\rangle, |B\rangle$ is the following: if we were to measure $|\psi\rangle$ in the computational basis, we would obtain $|k\rangle$ with probability $1/N$ and with probability $1 - 1/N$ obtain an i for which $x_i = 0$. Grover's algorithm consists

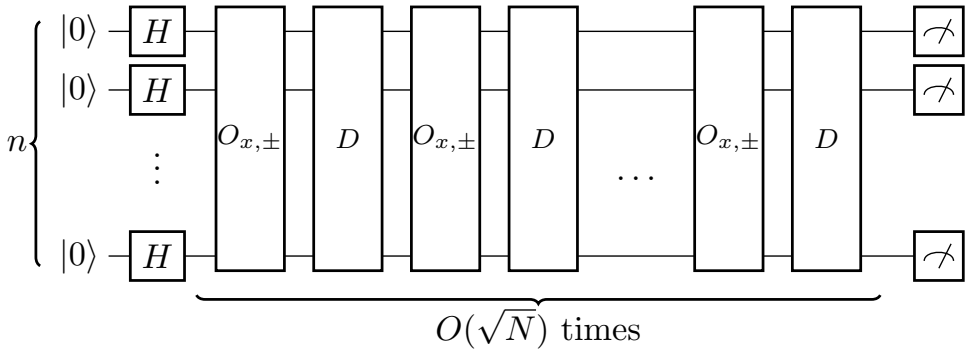


Figure 2.6: Grover's algorithm. Begin with the all-0 state and apply $H^{\otimes n}$ to create the uniform superposition state. Then apply $(O_{x,\pm} \cdot D)$ $O(\sqrt{N})$ many times and measure the outcome. The final measurement gives us the solution k with high probability (using an additional classical query, one can easily check if $x_k = 1$).

of a sequence of $O(\sqrt{N})$ applications of $DO_{x,\pm}$ that drives the amplitude of $|G\rangle$ from $\frac{1}{\sqrt{N}}$ to approximately 1 so that a final measurement in Fig. 2.6 yields k with high probability.

We now explain one application of $DO_{x,\pm}$ in a geometric manner. Since $|G\rangle$ and $|B\rangle$ are orthogonal states, let us work in the 2-dimensional space spanned by the states $|G\rangle$ and $|B\rangle$. In this direction, let θ be such that $\sin(\theta) = \frac{1}{\sqrt{N}}$. Then, from Eq. (2.4), we have $|\psi\rangle = \sin(\theta)|G\rangle + \cos(\theta)|B\rangle$. After the first query $O_{x,\pm}$, the state of the algorithm can be written as

$$\frac{1}{\sqrt{N}} \sum_{i \in \{0,1\}^n} (-1)^{x_i} |i\rangle = -\sin(\theta)|G\rangle + \cos(\theta)|B\rangle,$$

which is effectively a reflection around the basis state $|B\rangle$ (see the second diagram in Fig. 2.7 for a geometric perspective). In order to understand the action of $D = (2|\psi\rangle\langle\psi| - \mathbf{1}_N)$, let $|\phi\rangle$ to be a state (in the span of $\{|G\rangle, |B\rangle\}$) that is *orthogonal* to $|\psi\rangle$. Then,

$$D|\psi\rangle = |\psi\rangle, \qquad D|\phi\rangle = -|\phi\rangle,$$

hence D implements a reflection around the state $|\psi\rangle$. As is well-known in linear algebra, the product of two reflections amounts to a rotation, so the action of $(D \cdot O_{x,\pm})$ on $|\psi\rangle$ can be seen as a rotation towards the good state $|G\rangle$. This is illustrated in Fig. 2.7.

The action of $DO_{x,\pm}$ on $|\psi\rangle$ can be written as

$$DO_{x,\pm} : \sin(\theta)|G\rangle + \cos(\theta)|B\rangle \to \sin(3\theta)|G\rangle + \cos(3\theta)|B\rangle.$$

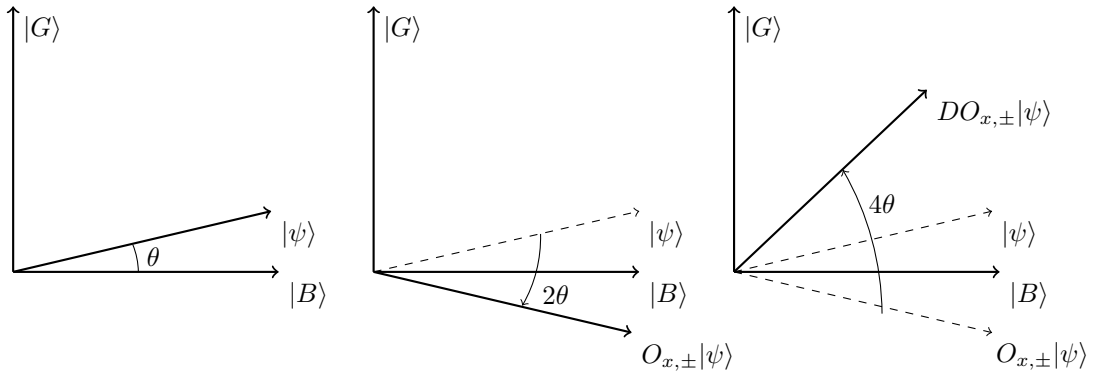


Figure 2.7: The first figure expresses $|\psi\rangle = \sin(\theta)|G\rangle + \cos(\theta)|B\rangle$ in the $\{|G\rangle, |B\rangle\}$ basis. The action of $DO_{x,\pm}$ on $|\psi\rangle$ is described in two stages: first $O_{x,\pm}$ performs a reflection through $|B\rangle$ (second figure), next D performs a reflection through the dashed $|\psi\rangle$ (third figure). The net effect is a rotation of $|\psi\rangle$ by an angle 2θ.

More generally, suppose we apply $DO_{x,\pm}$ t times to $|\psi\rangle$ (for some t that we pick later), we obtain

$$(DO_{x,\pm})^t : \sin(\theta)|G\rangle + \cos(\theta)|B\rangle \to \sin((2t+1)\theta)|G\rangle + \cos((2t+1)\theta)|B\rangle.$$

We now want to pick t so that $\sin^2((2t+1)\theta)$ is close to 1, so that a measurement of the final state would yield $|G\rangle$ with high probability.[7] Ideally, we should pick $t' = \frac{\pi}{4\theta} - 1/2$, so that $\sin^2((2t'+1)\theta) = 1$, but it is not clear if t' is even an integer. Instead, we pick t to be the largest integer less than $\frac{\pi}{4\theta} - 1/2$.

Let us analyze the success probability of the algorithm for this choice of t. First note that $|t - (\frac{\pi}{4\theta} - 1/2)| \leq 1$. Using this, observe that

$$\begin{aligned}\sin^2((2t+1)\theta) \geq \sin^2\left(\left(2\left(\frac{\pi}{4\theta} - \frac{3}{2}\right) + 1\right)\theta\right) &= \sin^2(\pi/2 - 2\theta) \\ &= 1 - \sin^2(2\theta) \geq 1 - 4/N,\end{aligned}$$

where the first inequality used the monotonicity of $\sin(\phi)$ for $\phi \in [0, \pi/2]$ and the last inequality used that $\sin(\theta) = 1/\sqrt{N}$.

Let us now analyze the query complexity and gate complexity of Grover's algorithm. Using $\arcsin(\phi) \geq \phi$ for $\phi > 0$, clearly $t \leq \frac{\pi}{4}\sqrt{N}$. So, the query complexity is at most $\frac{\pi}{4}\sqrt{N}$. In order to analyze the gate complexity, it remains to analyze the number of gates involved in implementing D (we do not count

[7] Note that we also need to pick t so that we do not "overshoot" $|G\rangle$, i.e., in Fig. 2.7 each application of $O_{x,\pm}D$ is a rotation by 2θ anti-clockwise, so for some t, $(DO_{x,\pm})^t|\psi\rangle$ will get close to $|G\rangle$ and for $q \geq t$, $(DO_{x,\pm})^q|\psi\rangle$ will cross over $|G\rangle$ and move to the second quadrant in the 2-dimensional space spanned by the states $|G\rangle$ and $|B\rangle$.

the application of a query $O_{x,\pm}$ as a gate). Let $D_n = 2|0^n\rangle\langle 0^n| - \mathbf{1}_N$ be the n-qubit unitary that reflects through $|0^n\rangle$. It is not hard to see that this can be implemented using $O(n)$ elementary gates and $n-1$ auxiliary qubits that all start and end in $|0\rangle$ (and that we often will not even write explicitly). Specifically, one can apply $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ gates to each of the n qubits, then use $n-1$ Toffoli gates into $n-1$ auxiliary qubits to compute the logical AND of the first n qubits, then apply $-Z$ to the last qubit (which negates the basis states where this AND is 0), and reverse the Toffolis and Xs. So, the overall gate complexity of implementing D_n is $4n-1$. Along with the n Hadamard gates applied at the start of the algorithm, the overall gate complexity of Grover's algorithm is $O(\sqrt{N}\log N)$.

This concludes the analysis of Grover's algorithm and in particular shows that we can solve unstructured search problem using $O(\sqrt{N})$ quantum queries and $O(\sqrt{N}\log N)$ elementary gates. Furthermore, we show in the next section that Grover's algorithm is in fact optimal in terms of queries, i.e., there exists no quantum algorithm that can search an unstructured database by making fewer queries.[8] In Chapter 3, we will present another search algorithm that has essentially the same query complexity of Grover's algorithm and has gate complexity $O(\sqrt{N}\log(\log^\star N))$.

2.5.2 Quantum lower bound for search

The first known query lower bound for quantum algorithms solving the search problem was shown before Grover's search algorithm was even discovered! Bennett et al. in 1993 [BBBV97] used the so-called hybrid-method to show that every quantum algorithm that solves the search problem requires $\Omega(\sqrt{N})$ queries. In this section, we do not describe their proof. Instead, we give two lower bound proofs for the search problem: using the polynomial method and the positive-weight adversary method.

Using polynomial method. A priori, it is not clear, how one would use Corollary 2.4.3 to show degree lower bounds for multilinear polynomials p that approximate a Boolean function (say up to error $1/3$). However, for the class of *symmetric* Boolean functions, the quest for approximate degree lower bounds can be highly simplified and the polynomial method seems like a natural approach to lower bound the quantum query complexity of such functions. A symmetric Boolean function $f : \{0,1\}^N \to \{0,1\}$ is a function whose value at $x \in \{0,1\}^N$ depends only on $|x|$, i.e., the Hamming weight of x. One such example is OR_N. Clearly when $|x| \geq 1$, we know $\mathrm{OR}_N(x) = 1$, and otherwise $\mathrm{OR}_N(x) = 0$. A general

[8] Zalka [Zal99] in fact showed that even the constant in Grover's algorithm is optimal, i.e., one cannot hope to solve unstructured search with fewer than $\frac{\pi}{4}\sqrt{N}$ quantum queries.

technique used in proving degree lower bounds for symmetric functions is to use *symmetrization* to convert multivariate polynomials to a *univariate* polynomial and then prove degree lower bounds for these simpler polynomials. Univariate polynomials have been studied in approximation theory for several decades, which allows us to borrow results from approximation theory to prove lower bounds on quantum query complexity.

Let p be a multilinear polynomial. So p can be written as a sum of monomials $p(x) = \sum_{S\subseteq[n]} c_S \prod_{i\in S} x_i$ for some $c_S \in \mathbb{R}$. The symmetrized multilinear polynomial p' associated with p is defined as

$$p'(x) = \frac{1}{N!} \sum_{\pi\in S_N} p(\pi(x)).$$

Observe that $\deg(p') \leq \deg(p)$. Now, one can show the existence of a univariate polynomial $P : \mathbb{R} \to \mathbb{R}$ defined as $P(|x|) = p'(x)$ for every $x \in \{0,1\}^N$ that satisfies $\deg(P) \leq \deg(p')$.[9] In order to see this, first note that by explicitly writing out every summand in $p'(x)$ in terms of the decomposition $p(\pi(x)) = \sum_{S\subseteq[n]} c_S \prod_{i\in S}(\pi(x))_i$, every monomial in p' with degree i has the same coefficient $\sum_{S:|S|=i} c_S$. Next, suppose $x \in \{0,1\}^N$ satisfies $|x| = k$, then exactly $\binom{k}{i}$ monomials of degree i evaluate to 1 and the rest evaluate to 0. Using these two observations we can write p' as $p'(x) = \sum_{i=0}^{N} \binom{|x|}{i}$. We now define P to be $P(k) = \sum_{i=0}^{N} \binom{k}{i}$ for every $k \in \{0,\ldots,N\}$ so that it satisfies $P(|x|) = p'(x)$. Since the binomial coefficient $\binom{k}{i}$ is a degree-i polynomial in k, it follows that $\deg(P) \leq \deg(p')$.

Let p be an N-variate polynomial that approximates OR_N up to error $1/3$, i.e., $|p(x) - \mathrm{OR}_N(x)| \leq 1/3$ for every $x \in \{0,1\}^N$. Define a univariate polynomial $P : \mathbb{R} \to \mathbb{R}$ as above. Abusing notation, let us also define $\mathrm{OR}_N(k) = \mathbb{E}_{|x|=k}[\mathrm{OR}_N(x)]$. Then, it follows that

$$|P(k) - \mathrm{OR}_N(k)| = \Big| \mathop{\mathbb{E}}_{\substack{x\in\{0,1\}^N:\\ |x|=k}} [p(x) - \mathrm{OR}_N(x)] \Big| \leq 1/3,$$

for every $k \in \{0,\ldots,N\}$. In particular, by the property of the OR_N function, P satisfies the following properties: (i) $P(0) \in [-1/3, 1/3]$ and (ii) $P(k) \in [2/3, 4/3]$ for every $k \in [N]$. Nisan and Szegedy [NS94] used a theorem of Ehlich and Zeller [EZ64] and Rivlin and Cheney [RC66], to show that *every* univariate polynomial P satisfying property (i) and (ii) must have degree $\Omega(\sqrt{N})$.[10], Using the

[9] Note that $\deg(P)$ refers to the degree of the univariate polynomial P and $\deg(p')$ refers to the degree of the multilinear polynomial p'.

[10] Nisan and Szegedy [NS94] also construct an explicit polynomial based on the Chebyshev polynomial that attains this degree bound. The existence of such a polynomial also follows from Grover's algorithm. Indeed, using Corollary 2.4.3, it follows that the acceptance probability of Grover's algorithm is a degree-$O(\sqrt{N})$ polynomial that approximates the OR_N function.

remark in the previous paragraph, it now follows that, every multilinear polynomial p satisfying $|p(x) - \mathrm{OR}_N(x)| \leq 1/3$ for every $x \in \{0,1\}^N$, must have degree $\Omega(\sqrt{N})$. Along with Corollary 2.4.3 (the polynomial method), we have that $Q(\mathrm{OR}_N) = \Omega(\sqrt{N})$, which in particular shows that Grover's search algorithm is optimal in terms of the number of queries.

Using the adversary method. In order to prove the lower bound for search, it suffices to look at the positive-weight adversary method. As discussed in Section 2.4.2, the crux in proving a good lower bound using this method is to cleverly define a relation $R \subseteq f^{-1}(0) \times f^{-1}(1)$ that maximizes $\sqrt{mm'/\ell\ell'}$. In order to do so, let

$$R = \{(0^N, e_1), (0^N, e_2), \ldots, (0^N, e_N)\} \subseteq \mathrm{OR}_N^{-1}(0) \times \mathrm{OR}_N^{-1}(1).$$

Note that $m = N$ and $m' = 1$ because the 0^N appears in every element of the relation and the e_is appear in exactly one element of the relation. Clearly, $\ell = \ell' = 1$. This gives an overall lower bound of $Q(f) = \Omega(\sqrt{N})$, yet again showing that Grover's algorithm is indeed optimal.

Chapter 3

Gate complexity of quantum search

This chapter is based on the paper "Optimizing the Number of Gates in Quantum Search", by S. Arunachalam and R. de Wolf [AW17c].

Abstract. In Chapter 2, we described Grover's search algorithm to find a solution in an N-bit database. The algorithm used $O(\sqrt{N})$ queries and $O(\sqrt{N}\log N)$ gates. Bennett et al. [BBBV97] showed that *every* search algorithm needs to make $\Omega(\sqrt{N})$ queries, so the quantum query complexity of the search problem is $\Theta(\sqrt{N})$. In this chapter we are concerned with the number of gates needed for quantum search algorithms. Grover in 2002 [Gro02] showed how to reduce the number of gates to $O(\sqrt{N}\log\log N)$ for the special case where the database has a unique solution, without significantly increasing the number of queries. We show how to reduce this further to $O(\sqrt{N}\log^{(r)} N)$ gates for every constant r, and sufficiently large N. This means that, on average, the circuits between two queries barely touch more than a constant number of the $\log N$ qubits on which the algorithm acts. For a very large N that is a power of 2, we can choose r such that the algorithm uses essentially the minimal number $\frac{\pi}{4}\sqrt{N}$ of queries, and only $O(\sqrt{N}\log(\log^{\star} N))$ other gates.

Contents

3.1 Introduction

In this chapter, we will focus on quantum algorithms for the *unstructured search* problem defined as follows:

3.1.1. Definition (*Unstructured search*). Let $n \in \mathbb{N}$ and $N = 2^n$. Given a database modelled as $x \in \{0,1\}^N$, the goal is to find an index $i \in [N]$ such that $x_i = 1$ (we refer to such an i as a *solution*) and output 'no solutions' if there exists no such i.

In order to solve this problem, we are allowed to make quantum queries, which correspond to the transformation

$$O_{x,\pm} : |i\rangle \to (-1)^{x_i}|i\rangle.$$

The goal is to find a solution, making as few queries as possible. The standard version of Grover's algorithm (discussed in Section 2.5.1) finds a solution with high probability using $O(\sqrt{N})$ quantum queries and $O(\sqrt{N}\log N)$ other elementary gates. The algorithm can be quickly summarized as follows: it starts from a uniform superposition over all database-indices i, and then applies $O(\sqrt{N})$ identical "iterations," each of which uses one query and $O(\log N)$ other elementary gates. Together these iterations concentrate most of the amplitude on the solution(s). A measurement of the final state then yields a solution with high probability. For the special case of a database with a unique solution its number of iterations (= number of queries) is essentially $\frac{\pi}{4}\sqrt{N}$, and Zalka [Zal99] showed that this number of queries is optimal.

In [Gro02], Grover gave an alternative algorithm to find a unique solution using slightly more (but still $(\frac{\pi}{4} + o(1))\sqrt{N}$) queries, and only $O(\sqrt{N}\log\log N)$ other elementary gates. The algorithm is more complicated than the standard Grover algorithm, and no longer consists of $O(\sqrt{N})$ identical iterations. Still, it acts on $O(\log N)$ qubits, so on average a unitary sitting between two queries acts on only a tiny $O(\log\log N/\log N)$-fraction of the qubits. It is quite surprising that such mostly-very-sparse unitaries suffice for quantum search.

In this chapter we show how Grover's reduction in the number of gates can be improved further: for every fixed r, and sufficiently large N, we give a quantum algorithm that finds a unique solution in a database of size N using $O(\sqrt{N})$ queries and $O(\sqrt{N}\log^{(r)} N)$ other elementary gates.[1] To be concrete about the latter, we assume that the set of elementary gates at our disposal is the Toffoli (controlled-controlled-NOT) gate, and all one-qubit unitary gates.

Organization. This chapter is organized as follows. In Section 3.2, we begin by giving a proof sketch of our gate-optimized construction. In Section 3.3, we

[1] The constant in the $O(\cdot)$ for the number of gates depends on r. The iterated binary logarithm is defined as $\log^{(s+1)} = \log \circ \log^{(s)}$, where $\log^{(0)}$ is the identity function.

analyze the query and gate complexity of the well-known amplitude amplification procedure. In Section 3.4.1, we prove a technical theorem that recovers the result of Grover [Gro02]. In Section 3.4.2, we prove our main result and present some questions for future work in Section 3.5.

3.2 Overview of the proof

Here we first give a sketch of our gate-optimized quantum algorithm. Our approach is recursive: we build a quantum search algorithm for a larger database by applying amplitude amplification on a search algorithm for a smaller database.[2] Let us sketch this in a bit more detail. Suppose we perform r recursions in such a way that the final algorithm (after r recursions) solves the search problem on an N-bit database and let N_1 (that depends on r) be the smallest database-size that we begin with. We consider an increasing sequence of database-sizes $N_1, \ldots, N_r = N$, where $N_{i+1} \approx 2^{\sqrt{N_i}}$ (of course, N needs to be sufficiently large for such a sequence to exist). The basic Grover algorithm $\mathcal{C}^{(1)}$ can search a database of size N_1 using

$$Q_1 = O(\sqrt{N_1}), \qquad E_1 = O(\sqrt{N_1}\log N_1)$$

queries and gates, respectively. We now use $\mathcal{C}^{(1)}$ to construct a search algorithm for a database with size N_2 as follows. Think of the N_2-sized database as consisting of N_2/N_1 N_1-sized databases; we can just pick one such (N_1)-sized database uniformly at random and use algorithm $\mathcal{C}^{(1)}$ to search for a solution in the (N_1)-sized database. Assuming the (N_2)-sized database had a unique solution, the probability that the random (N_1)-sized database picked contained the solution is N_1/N_2. We now use $O(\sqrt{N_2/N_1})$ rounds of amplitude amplification to boost (to 1) the N_1/N_2 probability that our randomly chosen (N_1)-sized database happened to contain the unique solution. Each round of amplitude amplification involves one application of the smaller algorithm $\mathcal{C}^{(1)}$, one application of its inverse, a reflection through the $\log N_2$-qubit all-0 state, and one more query. This gives a search algorithm $\mathcal{C}^{(2)}$ for an N_2-sized database that uses

$$Q_2 = O\left(\sqrt{\frac{N_2}{N_1}}Q_1\right) = O(\sqrt{N_2}), \qquad E_2 = O\left(\sqrt{\frac{N_2}{N_1}}(E_1 + \log N_2)\right)$$

queries and gates respectively. Note that by our choice of $N_2 \approx 2^{\sqrt{N_1}}$, we have $E_1 = O(\sqrt{N_1}\log N_1) \geq \log N_2$, so $E_2 = O(\sqrt{N_2/N_1}E_1)$. The same approach

[2] The idea of doing recursive applications of amplitude amplification to search increasingly larger database-sizes is reminiscent of the algorithm of Aaronson and Ambainis [AA05] for searching an N-element database that is arranged in a d-dimensional grid. However, their goal was to design a search algorithm for the grid with nearest-neighbor gates and with optimal number of *queries* (they succeeded for $d > 2$). It was not to optimize the number of *gates*. If one writes out their algorithm as a quantum circuit, it still has roughly $\sqrt{N}\log N$ gates.

as above allows us to use $\mathcal{C}^{(2)}$ to construct a search algorithm for a database of size N_3. Repeating this construction iteratively gives a recursion

$$Q_{i+1} = O\left(\sqrt{\frac{N_{i+1}}{N_i}}Q_i\right), \qquad E_{i+1} = O\left(\sqrt{\frac{N_{i+1}}{N_i}}E_i\right).$$

The constant factor in the $O(\cdot)$ blows up by a constant factor in each recursion, so after r steps this unfolds to

$$Q_r = O(\exp(r)\sqrt{N}), \qquad E_r = O(\exp(r)\sqrt{N}\log N_1).$$

Since $N_1, \ldots, N_r = N$ is (essentially) an exponentially increasing sequence, we have $\log N_1 = O(\log^{(r)} N)$.

The result we prove in this chapter is stronger: it does not have the $\exp(r)$ factor. Tweaking the above idea to avoid this $\exp(r)$ factor is somewhat delicate, and will take up the remainder of this chapter. In particular, in order to get close to the optimal query complexity $\frac{\pi}{4}\sqrt{N}$, it is important (and different from Grover's approach) that the intermediate amplitude amplification steps do *not* boost the success probability all the way to 1. The reason is that amplitude amplification is less efficient when boosting large success probabilities to 1 than when boosting small success probabilities to somewhat larger success probabilities. Our final algorithm will boost the success probability to 1 only at the very end, after all r recursion steps have been done. Because the calculations involved are quite fragile, and tripped us up multiple times, the proofs in the body of the chapter are given in much detail.

If N is a power of 2, then choosing $r = \log^\star N$ in our result and being careful about the constants,[3] we get an exact quantum algorithm for finding a unique solution using essentially the optimal $\frac{\pi}{4}\sqrt{N}$ queries, and $O(\sqrt{N}\log(\log^\star N))$ elementary gates. Note that our algorithm on average uses only $O(\log(\log^\star N))$ elementary gates in between two queries, which is barely more than constant. Once in a while a unitary acts on many more qubits, but the average is only $O(\log(\log^\star N))$.

Possible objections. To pre-empt the critical reader, let us mention two objections one may raise against the fine-grained optimization of the number of elementary gates that we do here. First, one query acts on $\log N$ qubits, and when itself implemented using elementary gates, any oracle that's worth its salt would require $\Omega(\log N)$ gates. Since $\Omega(\sqrt{N})$ queries are necessary, a fair way of counting would say that just the queries themselves already have "cost" $\Omega(\sqrt{N}\log N)$, rendering our (and Grover's [Gro02]) gate-optimizations moot. Second, to do exact amplitude amplification in our recursion steps, we allow infinite-precision single-qubit phase gates. This is not realistic, as in practice such gates would have to be approximated by more basic gates. Our reply to both would be: fair

[3] The function $\log^\star N$ is the number of times the binary logarithm must be iteratively applied to N to obtain a number that is at most 1: $\log^\star N = \min\{r \geq 0 : \log^{(r)} N \leq 1\}$.

enough, but we still find it quite surprising that query-efficient search algorithms only need to act on a near-constant number of qubits in between the queries on average. It is interesting that after nearly two decades of research on quantum search, the basic search algorithm can still be improved in some ways. It may even be possible to optimize our results further to use $O(\sqrt{N})$ elementary gates, which would be even more surprising.

3.3 Gate complexity of exact amplitude amplification

Amplitude amplification is a technique that can be used to efficiently boost quantum search algorithms with a known success probability to higher success probability. We will invoke the following theorem from [BHMT02] in the proof of our main theorem later.

3.3.1. THEOREM. *Let $N = 2^n$. Suppose there exists a unitary quantum algorithm $\mathcal{A}$ that uses Q queries and E gates and finds a solution in database $x \in \{0,1\}^N$ with known probability a, in the sense that measuring $\mathcal{A}|0^n\rangle$ yields a solution with probability exactly a. Let $a' \in [a,1]$ and $w = \lceil \frac{\arcsin(\sqrt{a'})}{2\arcsin(\sqrt{a})} - \frac{1}{2} \rceil$. Then there exists a quantum algorithm $\mathcal{B}$ that finds a solution with probability exactly a' using $w+1$ applications of algorithm $\mathcal{A}$, w applications of $\mathcal{A}^{-1}$, w additional queries, and $4w(n+2)$ additional elementary gates. In total, $\mathcal{B}$ uses $(2w+1)Q+w$ queries and $w(4n+2E+8)+E$ elementary gates.*

Proof. For the sake of completeness we present the construction of quantum algorithm $\mathcal{B}$. The idea is to lower the success probability from a in such a way that an integer number of rounds of amplitude amplification suffice to produce a solution with probability exactly a'.

Define $\theta = \frac{\arcsin(\sqrt{a'})}{2w+1}$ and $\tilde{a} = \sin^2(\theta)$, where w is defined in the theorem. Let $R_{\tilde{a}/a}$ be the one-qubit rotation that maps $|0\rangle \mapsto \sqrt{\tilde{a}/a}|0\rangle + \sqrt{1-\tilde{a}/a}|1\rangle$. Call an $(n+1)$-bit string (i,b) a "solution" if $x_i = 1$ *and* $b = 0$. Define the $(n+1)$-qubit unitary $O'_x = (\mathbf{1}_n \otimes XH)O_x(\mathbf{1}_n \otimes HX)$. It is easy to verify that O'_x puts a " $-$ " in front of the solutions (in the new sense of the word), and a " $+$ " in front of the non-solutions.

Let $\mathcal{A}' = \mathcal{A} \otimes R_{\tilde{a}/a}$, and define $|U\rangle = \mathcal{A}'|0^{n+1}\rangle$ to be the final state of this new algorithm. Let $|G\rangle$ be the normalized projection of $|U\rangle$ on the (new) solutions and $|B\rangle$ be the normalized projection of $|U\rangle$ on the (new) non-solutions. Measuring $|U\rangle$ results in a (new) solution with probability exactly $\sin^2(\theta)$, hence we can write

$$|U\rangle = \sin(\theta)|G\rangle + \cos(\theta)|B\rangle.$$

Define $\mathcal{Q} = \mathcal{A}' D_{n+1} (\mathcal{A}')^{-1} O'_x$. This is a product of two reflections in the plane spanned by $|G\rangle$ and $|B\rangle$: O'_x is a reflection through $|G\rangle$, and $\mathcal{A}' D_{n+1} (\mathcal{A}')^{-1} =$

$2|U\rangle\langle U| - I$ is a reflection through $|U\rangle$ (similar to the action of $DO_{x,\pm}$ in Grover's search algorithm, see Fig. 2.6). As is well known in the analysis of Grover's algorithm and amplitude amplification, the product of these two reflections rotates the state over an angle 2θ. Hence after applying the operator $\mathcal{Q}$ w times to $|U\rangle$ we have the state

$$\mathcal{Q}^w|U\rangle = \sin((2w+1)\theta)|G\rangle + \cos((2w+1)\theta)|B\rangle = \sqrt{a'}|G\rangle + \sqrt{1-a'}|B\rangle,$$

since $(2w+1)\theta = \arcsin(\sqrt{a'})$. Thus the algorithm $\mathcal{A}'$ can be boosted to success probability a' using an integer number of applications of $\mathcal{Q}$.

Our new algorithm $\mathcal{B}$ is now defined as $\mathcal{Q}^w\mathcal{A}'$. It acts on $n+1$ qubits (all initially $|0\rangle$) and maps

$$|0^{n+1}\rangle \mapsto \sqrt{a'}|G\rangle + \sqrt{1-a'}|B\rangle,$$

so it finds a solution with probability exactly a'. Algorithm $\mathcal{B}$ uses $w+1$ applications of algorithm $\mathcal{A}$ together with elementary gate $R_{\tilde{a}/a}$; w applications of $\mathcal{A}^{-1}$ together with $R^{-1}_{\tilde{a}/a}$; w applications of O'_x (each of which involves one query to x and two other elementary gates, counting XH as one gate); and w applications of D_{n+1}, each of which takes $4n+3$ elementary gates (for a proof of this, see last paragraph of Section 2.5.1). Hence the total number of queries that $\mathcal{B}$ makes is at most $(2w+1)Q + w$ and the total number of elementary gates used by $\mathcal{B}$ is at most $(2w+1)E + 4w(n+2)$. □

Using this theorem, the following remark and corollary follow readily.

3.3.2. Remark. A very simply algorithm to which we can apply this theorem is $\mathcal{A} = H^{\otimes n}$. If our $N = 2^n$-bit database has a unique solution, then the success probability is $a = 1/N$. Let $a' = 1/k$ for some integer $k \geq 2$. Then, Theorem 3.3.1 implies an algorithm $\mathcal{C}^{(1)}$ that finds a solution with probability exactly $1/k$ using w queries and at most $O(w \log N)$ other elementary gates, where

$$w = \left\lceil \frac{\arcsin(\sqrt{a'})}{2\arcsin(\sqrt{a})} - \frac{1}{2} \right\rceil \leq \left\lceil \frac{\sqrt{N}(1+1/k)}{2\sqrt{k}} - \frac{1}{2} \right\rceil. \tag{3.1}$$

The inequality above follows from $\arcsin(z) \geq z$ for all $z \geq 0$, and $\sin(\frac{1+1/k}{\sqrt{k}}) \geq \frac{1}{\sqrt{k}}$ since $\sin(z) \geq z - z^3/6$ for $z \geq 0$.

In order to amplify the probability of an algorithm from $1/k$ to 1 we use the following corollary.

3.3.3. Corollary. *Let $k \geq 2$, n be integers, $N = 2^n$. Suppose there exists a quantum algorithm $\mathcal{D}$ that finds a unique solution in an N-bit database with probability exactly $1/k$ using $Q \geq \sqrt{k}$ queries and E elementary gates. Then there exists a quantum algorithm that finds the unique solution with probability 1 using at most $\frac{\pi}{2}Q\sqrt{k}(1+\frac{2}{\sqrt{k}})^2$ queries and $O(\sqrt{k}(n+E))$ elementary gates.*

Proof. Applying Theorem 3.3.1 to algorithm $\mathcal{D}$ with $a = 1/k$ and $a' = 1$, we obtain an algorithm that succeeds with probability 1 using at most $w'(2Q+1)+Q$ queries and $O(w'(n+E))$ gates, where

$$w' = \left\lceil \frac{\arcsin(1)}{2\arcsin(1/\sqrt{k})} - \frac{1}{2} \right\rceil \leq \frac{\pi}{4}(\sqrt{k}+1),$$

using $\arcsin(x) \geq x$ for $x \geq 0$ and $\lceil \frac{\pi}{4}\sqrt{k} - \frac{1}{2} \rceil \leq \frac{\pi}{4}(\sqrt{k}+1)$. Hence, the total number of queries in this new algorithm is at most

$$\begin{aligned} \frac{\pi}{4}(\sqrt{k}+1)(2Q+1) + Q &= \frac{\pi}{2}Q(\sqrt{k}+1)\Big(1 + \frac{1}{2Q} + \frac{2}{\pi(\sqrt{k}+1)}\Big) \\ &\leq \frac{\pi}{2}Q(\sqrt{k}+1)\Big(1 + \frac{2}{\sqrt{k}}\Big) \\ &\leq \frac{\pi}{2}Q\sqrt{k}\Big(1 + \frac{2}{\sqrt{k}}\Big)^2, \end{aligned}$$

where we used $Q \geq \sqrt{k}$ and $\pi(\sqrt{k}+1) \geq 2\sqrt{k}$ in the first inequality. The total number of gates is $O(w'(n+E)) = O(\sqrt{k}(n+E))$. □

The following easy fact will be helpful to get rid of some of the ceilings that come from Theorem 3.3.1.

3.3.4. Fact. If $k \geq 2$ and $\alpha \geq k$, then $\lceil \frac{\alpha}{2}(1+\frac{1}{k}) - \frac{1}{2} \rceil \leq \frac{\alpha}{2}(1+\frac{2}{k})$.

3.3.5. Fact. If $k \geq 3$ and $i \geq 2$, then $(2i+8)\log k < k^{i+1}$.

Proof. Fixing $i = 2$, it is easy to see that $12 \log k < k^3$ for $k \geq 3$. Similarly, fix $k = 3$ and observe that $(2i+8)\log 3 < 3^{i+1}$ for all $i \geq 2$. This implies the result for all $k \geq 3$ and $i \geq 2$, because the right-hand side grows faster than the left-hand side in both i and k. □

3.4 Improving the gate complexity for quantum search

3.4.1 Reproving Grover's optimized construction

In this section we first prove a technical theorem which will be recursively applied in proving our main result. Using this theorem, we first recover Grover's gate-optimized construction [Gro02].

3.4.1. THEOREM. *Let $k \geq 4$, $n \geq m + 2\log k$ be integers, $M = 2^m$ and $N = 2^n$. Suppose there exists a quantum algorithm $\mathcal{G}$ that finds a unique solution in an M-bit database with a known success probability exactly $1/k$, using $Q \geq k+2$ queries and E other elementary gates. Then there exists a quantum algorithm that finds a unique solution in an N-bit database with probability exactly $1/k$, using Q' queries and E' other elementary gates where,*

$$Q' \leq Q\sqrt{N/M}(1+4/k), \qquad \frac{E}{(1+1/k^3)}\sqrt{\frac{N}{M}} \leq E' \leq (3n+E)\sqrt{\frac{N}{M}}(1+3/k).$$

Proof. Consider the following algorithm $\mathcal{A}$:

1. Start with $|0^n\rangle$.
2. Apply the Hadamard gate to the first $n-m$ qubits, leaving the last m qubits as $|0^m\rangle$. The resulting state has a uniform superposition over the first $n-m$ qubits $\frac{1}{\sqrt{N/M}}\sum_{y\in\{0,1\}^{n-m}}|y\rangle|0^m\rangle$.
3. Apply the unitary $\mathcal{G}$ to the last m qubits (using queries to x, with the first $n-m$ address bits fixed).

The final state of algorithm $\mathcal{A}$ is

$$(H^{\otimes(n-m)} \otimes \mathcal{G})|0^n\rangle = \frac{1}{\sqrt{N/M}} \sum_{y\in\{0,1\}^{n-m}} |y\rangle\mathcal{G}|0^m\rangle.$$

The state $|y\rangle\mathcal{G}|0^m\rangle$ depends on y, because here $\mathcal{G}$ restricts to the M-bit database that corresponds to the bits in x whose n-bit address starts with y. Let t be the n-bit address corresponding to the unique solution in the database $x \in \{0,1\}^N$. Then the probability of observing $|t_1 \ldots t_n\rangle$ in the state $|t_1 \ldots t_{n-m}\rangle\mathcal{G}|0^m\rangle$ is exactly $1/k$. Suppose $\sqrt{a}$ is the amplitude of t in the final state of algorithm $\mathcal{A}$, then we have that $a = \frac{M}{kN}$. The total number of queries made by algorithm $\mathcal{A}$ is Q (from Step 3) and the total number of elementary gates is $n-m+E$ (from Steps 2 and 3).

Applying Theorem 3.3.1 to algorithm $\mathcal{A}$ by choosing $a' = 1/k$, we obtain an algorithm $\mathcal{B}$ using at most $w(2Q+1)+Q$ queries and $w(4n+2E+8)+E$ gates (from Theorem 3.3.1), where

$$\begin{aligned} w = \left\lceil \frac{\arcsin(\sqrt{a'})}{2\arcsin(\sqrt{a})} - \frac{1}{2} \right\rceil &\leq \left\lceil \frac{\sqrt{1/k}(1+1/k)}{2\sqrt{a}} - \frac{1}{2} \right\rceil \\ &\leq \left\lceil \frac{\sqrt{N}(1+1/k)}{2\sqrt{M}} - \frac{1}{2} \right\rceil \leq \frac{\sqrt{N}(1+2/k)}{2\sqrt{M}}. \end{aligned}$$

The first inequality above uses $\sin(\frac{1+1/k}{\sqrt{k}}) \geq \frac{1}{\sqrt{k}}$ (since $\sin(z) \geq z - z^3/6$ for $z \geq 0$), the second inequality follows from $\arcsin(z) \geq z$ (for $z \geq 0$) and the third

inequality uses Fact 3.3.4 (which we can apply because $\sqrt{N/M} = \sqrt{2^{n-m}} \geq \sqrt{2^{2\log k}} = k$ by the assumption of the theorem).

The total number of queries in algorithm $\mathcal{B}$ is at most

$$\begin{aligned} Q' = w(2Q+1) + Q &\leq Q\sqrt{N/M}(1+2/k) + \frac{1}{2}\sqrt{N/M}(1+2/k) + Q \\ &\leq Q\sqrt{N/M}(1+2/k) + \frac{Q}{2k}\sqrt{N/M} + \frac{Q}{k}\sqrt{N/M} \\ &\leq Q\sqrt{N/M}(1+4/k), \end{aligned}$$

where we used $Q \geq k+2$ and $\sqrt{N/M} \geq k \geq 4$ (by the assumption of the theorem) in the second inequality. Finally, the number of gates in $\mathcal{B}$ is

$$\begin{aligned} E' = w(4n+2E+8) + E &\leq \sqrt{N/M}(1+2/k)(2n+E+4) + E \\ &\leq (3n+E)\sqrt{N/M}(1+3/k), \end{aligned}$$

where we used $\sqrt{N/M} \geq 4$ in the second inequality.

It is not hard to see that the number of gates in $\mathcal{B}$ is at least

$$\begin{aligned} E' = w(4n+2E+8) + E \geq 2wE + E &= 2\left\lceil \frac{\arcsin(\sqrt{a'})}{2\arcsin(\sqrt{a})} - \frac{1}{2} \right\rceil E + E \\ &\geq 2\left(\frac{\sqrt{1/k}}{2(1+1/k^3)\sqrt{M/(kN)}} - \frac{1}{2} \right) E + E \\ &= \frac{E}{(1+1/k^3)}\sqrt{N/M}. \end{aligned}$$

The inequality follows from $\arcsin(\sqrt{a'}) \geq \sqrt{a'} = \sqrt{1/k}$ and

$$\arcsin(\sqrt{a}) = \arcsin\left(\sqrt{\frac{M}{kN}}\right) \leq \sqrt{\frac{M}{kN}}\left(1 + \frac{M}{kN}\right) \leq \sqrt{\frac{M}{kN}}\left(1 + \frac{1}{k^3}\right),$$

where the first inequality used $\arcsin(z) \leq z + z^3/2$ for all $z \in [0, 1/2]$ and the second inequality used $M/N \leq 1/k^2$ (by the assumption of the theorem). □

Suppose we now apply Theorem 3.3.1 once, to an algorithm that finds the unique solution in an M-bit database with probability $1/\log\log N$, we then get the following corollary, which was essentially the main result of Grover [Gro02].

3.4.2. Corollary. *Let $n \geq 25$ and $N = 2^n$. There exists a quantum algorithm that finds a unique solution in a database of size N with probability 1, using at most $(\frac{\pi}{4} + o(1))\sqrt{N}$ queries and $O(\sqrt{N}\log\log N)$ other elementary gates.*

Proof. Let $m = \lceil \log(n^2k^3) \rceil$ and $k = \log\log N$. Let $\mathcal{C}^{(1)}$ be the algorithm (described in Remark 3.3.2) on an M-bit database with $M = 2^m$ that finds the solution with probability exactly $1/k$. The query and gate complexity of $\mathcal{C}^{(1)}$ are

$$Q_1 = \left\lceil \frac{\arcsin(1/\sqrt{k})}{2\arcsin(1/\sqrt{M})} - \frac{1}{2} \right\rceil, \quad E_1 \leq \left\lceil \frac{\arcsin(1/\sqrt{k})}{2\arcsin(1/\sqrt{M})} - \frac{1}{2} \right\rceil \log M \tag{3.2}$$

respectively. In order to apply Theorem 3.4.1 using $\mathcal{C}^{(1)}$ as our base algorithm, it remains to verify $m + 2\log k \leq n$ and $Q_1 \geq k + 2$. Observe that $k \geq 4$ and

$$m + 2\log k \leq \log(2n^2k^5) = \log(2n^2 \log^5 n) \leq n,$$

where the last inequality is true for $n \geq 25$. We now lower bound Q_1,

$$Q_1 = \left\lceil \frac{\arcsin(1/\sqrt{k})}{2\arcsin(1/\sqrt{M})} - \frac{1}{2} \right\rceil \geq \frac{1/\sqrt{k}}{2\arcsin(1/(nk^{3/2}))} - 1 \geq 2k - 1. \tag{3.3}$$

The first inequality uses $\arcsin(x) \geq x$ for $x \geq 0$ in the numerator and $\arcsin(x)$ is an increasing function in $x \in [0,1]$ in the denominator (since $M = 2^m \geq 2^{\log n^2k^3} = n^2k^3$). The second inequality uses $\arcsin(z) \leq z + z^3/2$ for $z \in [0, 1/2]$ to conclude $\arcsin(1/(nk^{3/2})) \leq 1/(4k^{3/2})$.

Hence we can apply Theorem 3.4.1 using $\mathcal{C}^{(1)}$ as our base algorithm. This gives an algorithm $\mathcal{C}^{(2)}$ that finds the solution with probability exactly $1/k$. The total number of queries Q_2, made by algorithm $\mathcal{C}^{(2)}$ is

$$Q_2 = \underbrace{\left\lceil \frac{\sqrt{M}(1+1/k)}{2\sqrt{k}} - \frac{1}{2} \right\rceil}_{\text{upper bound on } Q_1 \text{ in Eq. 3.2}} \cdot \underbrace{\left(\sqrt{N/M}(1+4/k)\right)}_{\text{contribution from Theorem 3.4.1}}.$$

This in turn can be upper bounded by

$$Q_2 \leq \frac{\sqrt{M}(1+2/k)}{2\sqrt{k}} \sqrt{N/M}(1+4/k) \leq \sqrt{\frac{N}{4k}}(1+4/k)^2, \tag{3.4}$$

where the inequality follows from Fact 3.3.4 (since $m \geq 4\log k$). Using Theorem 3.4.1, the total number of gates in $\mathcal{C}^{(2)}$ is

$$E_2 = O\Big(\Big(3n + \underbrace{\left\lceil \frac{\sqrt{M}(1+\frac{1}{k})}{2\sqrt{k}} - \frac{1}{2} \right\rceil \log M}_{\text{upper bound on } E_1 \text{ in Eq. 3.2}}\Big)\sqrt{\frac{N}{M}}(1+\frac{3}{k})\Big).$$

This in turn can be upper bounded by

$$\begin{aligned} E_2 &\leq O\Big(\sqrt{\frac{N}{k}}\Big(\frac{3n\sqrt{k}(1+3/k)}{\sqrt{M}} + (1+3/k)^2 \log M\Big)\Big) \\ &\leq O\Big(\sqrt{\frac{N}{k}}\Big(\frac{3}{k} + (1+3/k)^2 \log M\Big)\Big) \\ &\leq O\Big(\sqrt{\frac{N}{k}}\Big(1+\frac{3}{k}\Big)^3 \log\log N\Big), \end{aligned} \tag{3.5}$$

where we used Fact 3.3.4 in the first inequality, $n\sqrt{k}(1+3/k) \leq \sqrt{M}/k$ (since $m \geq \log(n^2k^3)$) in the second inequality and $\log M = O(\log\log N)$ in the last inequality. Applying Corollary 3.3.3 to algorithm $\mathcal{C}^{(2)}$, we obtain an algorithm that succeeds with probability 1 using at most

$$\underbrace{\left(\sqrt{\frac{N}{4k}}(1+\frac{4}{k})^2\right)}_{\text{upper bound on } Q_2 \text{ in Eq. 3.4}} \cdot \underbrace{\frac{\pi}{2}\left(\sqrt{k}(1+\frac{2}{\sqrt{k}})^2\right)}_{\text{contribution from Corollary 3.3.3}} \leq \frac{\pi}{4}\sqrt{N}\left(1+\frac{4}{\sqrt{k}}\right)^4 \tag{3.6}$$

queries and

$$O\left(n\sqrt{k}+\sqrt{N}\left(1+\frac{3}{k}\right)^3\log\log N\right) \leq O\left(\sqrt{N}\left(1+\frac{3}{k}\right)^3\log\log N\right) \tag{3.7}$$

gates, where the inequality follows from $n\sqrt{k} = n\sqrt{\log\log N} \leq \sqrt{N}\log\log N$ (which is true for $n \geq 25$). Since $k = \log\log N$, it follows that the query complexity of the final algorithm (given by Eq. (3.6)) can be upper bounded by $(\frac{\pi}{4} + o(1))\sqrt{N}$ and the gate complexity (given by Eq. (3.7)) is at most $O(\sqrt{N}\log\log N)$. □

3.4.2 Main theorem

We now prove our main theorem (the claim in the abstract will be a corollary of this). Suppose, we use Theorem 3.4.1 recursively by starting from the improved algorithm in Corollary 3.4.2. This gives query complexity $O(\sqrt{N})$ and gate complexity $O(\sqrt{N}\log\log\log N)$. Doing this multiple times and being careful about the constant (which grows in each step of the recursion), we obtain the following result:

3.4.3. Theorem. *Let k be a power of 2 and $N \geq 2^{16}$ a sufficiently large power of 2. For every $r \in [\log^\star N]$, $k \in \{\log^\star N, \ldots, \lceil \log\log N \rceil\}$, there exists a quantum algorithm that finds a unique solution in a database of size N with probability exactly $1/k$, using at most*

$$\sqrt{\frac{N}{4k}}(1+4/k)^r \text{ queries and}$$
$$O\left(\sqrt{\frac{N}{k}}(1+6/k)^{2r-1}\max\{\log k, \log^{(r)} N\}\right) \text{ other elementary gates.}$$

Proof. We begin by defining a sequence of integers $n_1, \ldots, n_r$ satisfying $n_r = \log N$ and $n_{i-1} = \max\{(2i+6)\log k, \lceil \log(n_i^2 k^3)\rceil\}$ for $i \in \{2, \ldots, r\}$. Note that N needs to be sufficiently large for such a sequence to exist and we assume this in the theorem. Also, observe that $n_1 \geq 10\log k \geq 20$ (since $k \geq \log^\star N \geq 4$). We first prove the following claim about this sequence.

3.4.4. Claim. *If $i \in \{2, \ldots, r\}$, then $n_{i-1} + 2\log k \leq n_i$.*

Proof. We prove this claim using downward induction on i.

Base case. For the base case $i = r$, we have $n_r = \log N$. Note that $(2r + 6)\log k \leq \lceil \log(n_r^2 k^3) \rceil$ for sufficiently large N and $k \leq \log\log N$, hence $n_{r-1} = \max\{(2r+6)\log k, \lceil \log(n_r^2 k^3)\rceil\} = \lceil \log(n_r^2 k^3)\rceil$. Using this, it now follows that

$$\begin{aligned} n_{r-1} + 2\log k = \lceil \log(n_r^2 k^3)\rceil + 2\log k &\leq \log(2n_r^2 k^5) \\ &\leq \log(2\log^2 N \log^5 n) \leq \log N = n_r, \end{aligned}$$

where the last inequality assumed N is sufficiently large.

Induction hypothesis. Assume that we have $n_i + 2\log k \leq n_{i+1}$ for every $i \in \{j, \ldots, r\}$.

Induction step. We now prove $n_{j-1} + 2\log k \leq n_j$ by considering the two possible values for n_{j-1}.

Case 1. $n_{j-1} = (2j+6)\log k$. Then we have

$$n_{j-1} + 2\log k = (2j+8)\log k \leq \max\{(2j+8)\log k, \lceil \log(n_{j+1}^2 k^3)\rceil\} = n_j.$$

Case 2. $n_{j-1} = \lceil \log(n_j^2 k^3)\rceil$. We first show $n_{j-1} \leq n_j$:

$$n_{j-1} \leq \lceil \log(n_{j+1}^2 k^3)\rceil \begin{cases} \leq (2j+8)\log k = n_j & \text{if } n_j = (2j+8)\log k \\ = n_j & \text{if } n_j = \lceil \log(n_{j+1}^2 k^3)\rceil, \end{cases}$$

where the first inequality uses the induction hypothesis and the second inequality uses $n_j = \max\{(2j+8)\log k, \lceil \log(n_{j+1}^2 k^3)\rceil\}$. We can now conclude the inductive step:

$$\begin{aligned} n_{j-1} + 2\log k \leq \log(2n_j^2 k^5) &= 1 + 2\log n_j + 5\log k \\ &\leq n_j/2 + 5\log k \leq n_j/2 + n_j/2 = n_j. \end{aligned}$$

In the first inequality above we use $n_{j-1} \leq \log(2n_j^2 k^3)$, in the second inequality we use $n_j \geq n_1 \geq 10\log k \geq 20$ (since $n_{j-1} \leq n_j$ for $j \in \{2, \ldots, r\}$ and $k \geq 4$) to conclude $1 + 2\log n_j \leq n_j/2$ (which is true for $n_j \geq 20$) and in the last inequality we use $5\log k \leq n_j/2$. □

Using the sequence $n_1, \ldots, n_r$, we consider r database-sizes $2^{n_1} = N_1 \leq 2^{n_2} = N_2 \leq \cdots \leq 2^{n_r} = N_r = N$. For each $i \in [r]$, we will construct a quantum algorithm $\mathcal{C}^{(i)}$ on a database of size N_i that finds a unique solution with probability exactly $1/k$. Let Q_i and E_i be the query complexity and gate complexity, respectively, of algorithm $\mathcal{C}^{(i)}$. We have already constructed the required algorithm $\mathcal{C}^{(1)}$ (described in Remark 3.3.2) on an N_1-bit database using

$$Q_1 \leq \left\lceil \frac{\sqrt{N_1}(1 + 1/k)}{2\sqrt{k}} - \frac{1}{2} \right\rceil \leq \frac{\sqrt{N_1}(1 + 2/k)}{2\sqrt{k}}$$

queries, where the inequality follows from Fact 3.3.4 (since $N_1 \geq k^{10}$). Also, a similar argument as in Eq. (3.3) shows that

$$Q_1 \geq \frac{\sqrt{N_1}(1+1/k)}{2\sqrt{k}} - 1 \geq k+2,$$

where the first inequality uses $N_1 \geq k^{10}$, and the second inequality uses $k \geq 4$. Using Theorem 3.3.1, the number of gates E_1 used by $\mathcal{C}^{(1)}$ is

$$\begin{aligned}
&\left\lceil \frac{\sqrt{N_1}(1+1/k)}{2\sqrt{k}} - \frac{1}{2} \right\rceil (6\log N_1 + 8) + \log N_1 \\
&\leq \frac{\sqrt{N_1}(1+2/k)}{\sqrt{k}} (3\log N_1 + 4) + \log N_1 \\
&\leq \frac{4\sqrt{N_1}(1+2/k)}{\sqrt{k}} \log N_1 + \log N_1 \\
&\leq \frac{4\sqrt{N_1}(1+3/k)}{\sqrt{k}} \log N_1,
\end{aligned}$$

where we use Fact 3.3.4 (since $N_1 \geq k^{10}$) in the first inequality and $N_1 \geq k^{10}$ in the second and third inequality. It is not hard to see that the number of gates E_1 used by $\mathcal{C}^{(1)}$ is at least $E_1 \geq \sqrt{N_1/k}$.

For $i \in \{2, \ldots, r\}$, we apply Theorem 3.4.1 using $\mathcal{C}^{(i-1)}$ as the base algorithm and we obtain an algorithm $\mathcal{C}^{(i)}$ that succeeds with probability exactly $1/k$ (since $\mathcal{C}^{(1)}$ had success probability exactly $1/k$, every algorithm obtained by iteratively applying Theorem 3.4.1 also has success probability exactly $1/k$). We showed earlier in Claim 3.4.4 that $n_{i-1} + 2\log k \leq n_i$ and it also follows that $k+2 \leq Q_1 \leq \cdots \leq Q_r$ (since the database-sizes $N_1, \ldots, N_r$ are non-decreasing). Hence both assumptions of Theorem 3.4.1 are satisfied. The total number of queries Q_i used by $\mathcal{C}^{(i)}$ is

$$Q_i \leq \sqrt{\frac{N_i}{N_{i-1}}} Q_{i-1} \left(1 + \frac{4}{k}\right). \tag{3.8}$$

In order to analyze the number of gates used by $\mathcal{C}^{(i)}$ we need the following two claims.

3.4.5. Claim. *$E_i \geq \frac{1}{(1+1/k^3)^i} \sqrt{\frac{N_i}{k}}$ for all $i \in [r]$.*

Proof. The proof is by induction on i. For the base case, we observed earlier that $E_1 \geq \sqrt{N_1/k}$. For the induction step assume $E_{i-1} \geq \frac{1}{(1+1/k^3)^{i-1}} \sqrt{\frac{N_{i-1}}{k}}$. The claim follows immediately from the lower bound on E' in Theorem 3.4.1 since

$$E_i \geq \frac{E_{i-1}}{1+1/k^3} \sqrt{N_i/N_{i-1}} \geq \frac{1}{(1+1/k^3)^i} \sqrt{\frac{N_i}{k}}.$$

□

3.4.6. Claim. *Suppose* $n_1 = \lceil \log(n_2^2 k^3) \rceil$. *Then* $n_{i-1} = \lceil \log(n_i^2 k^3) \rceil$ *for all* $i \in \{2, \ldots, r\}$.

Proof. We prove the claim by induction on i. The base case $i = 2$ is the assumption of the claim.

For the inductive step, assume $n_{i-1} = \lceil \log(n_i^2 k^3) \rceil$ for some $i \geq 2$. Using this, it follows that

$$\log(n_i^2 k^4) \geq \lceil \log(n_i^2 k^3) \rceil \geq (2i+6) \log k = \log(k^{2i+6}) \tag{3.9}$$

where the second inequality is because of the definition of $n_{i-1} = \max\{(2i+6)\log k, \lceil \log(n_i^2 k^3) \rceil\}$. Hence, Eq. (3.9) implies

$$n_i \geq k^{i+1} > (2i+8) \log k$$

using Fact 3.3.5 ($k \geq 3$ and $i \geq 2$ hold by the assumption of the theorem and claim respectively). In particular, this implies $n_i = \max\{(2i+8)\log k, \lceil \log(n_{i+1}^2 k^3) \rceil\}$ must be equal to the second term in the max. This concludes the proof of the inductive step and hence of the claim. □

Recursively it follows that the number of gates E_i used by $\mathcal{C}^{(i)}$ is at most

$$\begin{aligned}
\sqrt{\frac{N_i}{N_{i-1}}}(E_{i-1} + 3n_i)\Big(1 + \frac{3}{k}\Big) &= \sqrt{\frac{N_i}{N_{i-1}}} E_{i-1}\Big(1 + \frac{3n_i}{E_{i-1}}\Big)\Big(1 + \frac{3}{k}\Big) \\
&\leq \sqrt{\frac{N_i}{N_{i-1}}} E_{i-1}\Big(1 + 3n_i\Big(1 + \frac{1}{k^3}\Big)^{i-1}\sqrt{\frac{k}{N_{i-1}}}\Big)\Big(1 + \frac{3}{k}\Big) \\
&\leq \sqrt{\frac{N_i}{N_{i-1}}} E_{i-1}\Big(1 + \frac{3}{k}\Big(1 + \frac{1}{k^3}\Big)^{i-1}\Big)\Big(1 + \frac{3}{k}\Big) \\
&\leq \sqrt{\frac{N_i}{N_{i-1}}} E_{i-1}\Big(1 + \frac{6}{k}\Big)^2,
\end{aligned} \tag{3.10}$$

where we used Claim 3.4.5 in the first inequality to lower bound E_{i-1}, $n_i \leq \sqrt{\frac{N_{i-1}}{k^3}}$ in the second inequality (which clearly holds if $n_{i-1} = (2i+6)\log k \geq \lceil \log(n_i^2 k^3) \rceil$) and in the last inequality we used

$$\Big(1 + \frac{1}{k^3}\Big)^{i-1} \leq e^{(i-1)/k^3} \leq e^{r/k^3} \leq e^{1/(\log^\star N)^2} \leq 2,$$

since $r \leq \log^\star N$ and $k \geq \log^\star N$. Unfolding the recursion in Equations (3.8) and (3.10), we obtain

$$Q_r \leq \sqrt{\frac{N_r}{4k}}\Big(1 + \frac{4}{k}\Big)^r, \qquad E_r \leq 4\sqrt{\frac{N_r}{k}}\Big(1 + \frac{6}{k}\Big)^{2r-1} \log N_1.$$

It remains to show that $\log N_1 = n_1$, defined as $\max\{10\log k, \lceil\log(n_2^2 k^3)\rceil\}$, is in fact $O(\max\{\log k, \log^{(r)} N\})$. If $n_1 = 10\log k$, then we are done. If $n_1 = \lceil\log(n_2^2 k^3)\rceil$, we can use Claim 3.4.6 to write

$$n_{i-1} = \lceil 2\log n_i + 3\log k\rceil \le 4\log n_i, \quad \text{for } i \in \{2,\dots,r\},$$

where the last inequality follows from $k \le n_2^{1/3} \le n_i^{1/3}$ (using $\lceil\log(n_2^2k^3)\rceil \ge 10\log k$ for the first inequality and Claim 3.4.4 for the second inequality). Since $n_r = \log N$, it follows easily that $n_1 = O(\log^{(r)} N)$.
We conclude $n_1 = O(\max\{\log k, \log^{(r)} N\})$. □

The following is our main result:

3.4.7. Corollary.

- *For every constant integer $r > 0$ and sufficiently large $N = 2^n$, there exist a quantum algorithm that finds a unique solution in a database of size N with probability 1, using $(\frac{\pi}{4} + o(1))\sqrt{N}$ queries and $O(\sqrt{N}\log^{(r)} N)$ gates,*
- *For every $\varepsilon > 0$ and sufficiently large $N = 2^n$, there exist a quantum algorithm that finds a unique solution in a database of size N with probability 1, using $(\frac{\pi}{4} + \varepsilon)\sqrt{N}$ queries and $O(\sqrt{N}\log(\log^\star N))$ gates.*

Proof. Applying Corollary 3.3.3 to algorithm $\mathcal{C}^{(r)}$ (as described in Theorem 3.4.3), for some $k \le \log\log N$ to be specified later, we obtain an algorithm that succeeds with probability 1 using at most

$$\underbrace{\left(\sqrt{\frac{N}{4k}}\left(1+\frac{4}{k}\right)^r\right)}_{\text{upper bound on } Q_r \text{ from Theorem 3.4.3}} \cdot \underbrace{\frac{\pi}{2}\left(\sqrt{k}\left(1+\frac{2}{\sqrt{k}}\right)^2\right)}_{\text{contribution from Corollary 3.3.3}} \le \frac{\pi}{4}\sqrt{N}\left(1+\frac{4}{\sqrt{k}}\right)^{r+2}$$

queries and

$$O\left(\sqrt{k}n + \sqrt{N}\left(1+\frac{6}{k}\right)^{2r-1}\max\{\log k, \log^{(r)} N\}\right)$$
$$\le O\left(\sqrt{N}\left(1+\frac{6}{k}\right)^{2r}\max\{\log k, \log^{(r)} N\}\right)$$

gates. To obtain the two claims of the corollary we can now either pick:

- constant $r > 0$ and $k = (c_1\log^\star N)^2$, where $c_1 \in [1,2]$ is chosen to ensure k is a power of 2. It follows that

  $$\left(1+\frac{4}{\sqrt{k}}\right)^{r+2} = \left(1+\frac{4}{c_1\log^\star N}\right)^{r+2} = 1+o(1)$$

  for constant r. Similarly, $(1+6/k)^{2r} = 1+o(1)$. Since $\log^\star N \in o(\log^{(r)} N)$ for every constant r, we have $\max\{\log k, \log^{(r)} N\} = \log^{(r)} N$. Hence, the query and gate complexities are $(\frac{\pi}{4}+o(1))\sqrt{N}$ and $O(\sqrt{N}\log^{(r)} N)$, respectively.

- $r = \log^\star N$ and $k = (c_2(\log^\star N + 2))^2$, where we choose c_2 to be the smallest number that is at least $4/\ln(1+\varepsilon)$ and that makes k a power of 2. We have

$$\left(1 + \frac{4}{\sqrt{k}}\right)^{r+2} = \left(1 + \frac{4}{c_2(\log^\star N + 2)}\right)^{\log^\star N + 2} \leq 1 + \varepsilon,$$

  where the inequality used $(1+x)^y \leq e^{xy}$. Like before, it follows that $(1 + 6/k)^{2r} = 1 + o(1)$. Hence, the query and gate complexities are $\frac{\pi}{4}\sqrt{N}(1+\varepsilon)$ and $O(\sqrt{N}\log(\log^\star N))$, respectively.

□

3.5 Conclusion and future work

In this chapter, we constructed a new quantum algorithm that improves upon Grover's search algorithm in terms of gate complexity. In particular, our quantum algorithm finds the unique solution in an N-bit database using $O(\sqrt{N})$ database queries and $O(\sqrt{N}\log(\log^\star N))$ elementary gates.

Our work could be improved further in a number of directions:

- Can we remove the $\log(\log^\star N)$ factor in the gate complexity, reducing this to the optimal $O(\sqrt{N})$? This may well be possible, but requires a different idea than our roughly $\log^\star$ recursion steps, which will inevitably end up with $\omega(\sqrt{N})$ gates.

- Our construction only works for specific values of N. Can we generalize it to work for all sufficiently large N, even those that are not powers of 2, while still using close to the optimal $\frac{\pi}{4}\sqrt{N}$ queries?

- Can we obtain a similar gate-optimized construction when the database has *multiple* solutions instead of one unique one? Say when the exact number of solutions is known in advance?

- Most applications of Grover's algorithm deal with databases with an unknown number of solutions, and focus only on the number of queries. Are there application where our reduction in the number of elementary gates for search with one unique solution is both applicable and significant?

Chapter 4

Refining the polynomial method

This chapter is based on the paper "Quantum query algorithms are completely bounded forms", by S. Arunachalam, J. Briët and C. Palazuelos [ABP18]

Abstract. In Chapter 2, we discussed the polynomial method introduced by Beals et al. [BBC+01] in 1998 to give lower bounds on quantum query complexity. The polynomial method still remains one of the best-known lower-bound techniques for quantum query complexity. In this chapter, we refine this method by providing a characterization of quantum query algorithms in terms of polynomials satisfying a certain (completely bounded) norm constraint. Based on this characterization, we obtain a refined notion of approximate polynomial degree that equals the quantum query complexity, answering an open question of Aaronson et al. [AAI+16]. Using this characterization, we show that most polynomials of degree at least 4 are far from those coming from quantum query algorithms. We also give a simple and short proof of one of the results of Aaronson et al. showing a surprising equivalence between one-query quantum algorithms and bounded quadratic polynomials.

Contents

4.1 Introduction

Consider a function $f : D \to \{-1, 1\}$ on the domain $D \subseteq \{-1, 1\}^n$. In the black-box model of computation, promised that $x \in D$, the goal is to learn $f(x)$, when only given access to x through the oracle. An application of the oracle is usually referred to as a *query*. The bounded-error quantum query complexity of f, denoted $Q_\varepsilon(f)$, is the minimal number of queries a quantum algorithm must make on the worst-case input $x \in D$ to compute $f(x)$ with probability at least $1 - \varepsilon$, where $\varepsilon \in [0, 1/2)$ is usually some fixed but arbitrary positive constant.

For a detailed introduction to quantum query complexity, we refer the reader to Chapter 2. There, we also discussed the polynomial method and the adversary method to give lower bounds on quantum query complexity. In particular, it is known that the "negative-weight" adversary method characterizes quantum query complexity. However, proving lower bounds using negative-weight adversary method appears to be hard in general. In this chapter, our focus will be on *characterizing* quantum query complexity from the perspective of polynomials, which was not known before, to the best of our knowledge.

4.1.1 The polynomial method

The polynomial method is based on a connection between quantum query algorithms and polynomials discovered by Beals et al. [BBC+01]. They observed that for every t-query quantum algorithm $\mathcal{A}$ that on input $x \in \{-1, 1\}^n$ returns a random sign $\mathcal{A}(x)$, there exists a degree-$(2t)$ polynomial p such that $p(x) = \mathbb{E}[\mathcal{A}(x)]$ for every x (where the expectation is taken over the randomness of the output). It follows that if $\mathcal{A}$ computes $f : \{-1, 1\}^n \to \{-1, 1\}$ with probability at least $1 - \varepsilon$, then p satisfies $|p(x) - f(x)| \leq 2\varepsilon$ for every $x \in D$ and satisfies $|p(x)| \leq 1$ for all x.[1] The polynomial method thus converts the problem of lower bounding

[1] In Section 2.4.1, we showed that if $\mathcal{A}$ computes $f : \{-1, 1\}^n \to \{0, 1\}$ with error probability $\leq \varepsilon$ using t queries, then the probability that $\mathcal{A}$ outputs 1 after t queries, on input x, is given by a degree-$(2t)$ polynomial $q(x)$, which satisfies $|q(x) - f(x)| \leq \varepsilon$. Let $f^{\pm}(x) = 1 - 2f(x)$ (in order to change the range of f from $\{0, 1\}$ to $\{1, -1\}$), suppose $\mathcal{A}^{\pm}$ outputs $\{1, -1\}$ instead

quantum query complexity to the problem of proving lower bounds on the minimum degree of a polynomial p such that $|p(x) - f(x)| \leq 2\varepsilon$ holds for all inputs x. The minimal degree of such a polynomial is called the *approximate (polynomial) degree* and is denoted by $\deg_\varepsilon(f)$.

Notable applications of this approach showed optimality for Grover's search algorithm [BBC+01][2] and collision-finding and element distinctness [AS04]. In a recent work, Bun et al. [BKT17] use the polynomial method to resolve the quantum query complexity of several other well-studied Boolean functions.

4.1.2 Converses to the polynomial method.

A natural question is whether the polynomial method admits a converse. If so, this would imply a succinct characterization of quantum algorithms in terms of basic mathematical objects. However, Ambainis [Amb06] answered this question in the negative, showing that for infinitely many n, there is a function f with $\deg_{1/3}(f) \leq n^\alpha$ and $Q_{1/3}(f) \geq n^\beta$ for some positive constants $\beta > \alpha$ (recently larger separations were obtained in [ABK16, BKT17]). The approximate degree thus turns out to be an imprecise measure for quantum query complexity in general. These negative results would still leave room for the following two possibilities:

1. There is a (simple) refinement of approximate polynomial degree that characterizes $Q_\varepsilon(f)$ up to a constant factor.

2. Constant-degree polynomials characterize constant-query quantum algorithms.

These avenues were recently explored by Aaronson and others [AA15, AAI+16]. The first work strengthened the polynomial method by observing that quantum algorithms give rise to polynomials with a so-called *block-multilinear* structure. Based on this observation, they introduced a refined degree measure, $\text{bm-deg}_\varepsilon(f)$ which lies between $\deg_\varepsilon(f)$ and $2Q_\varepsilon(f)$, prompting the immediate question of how well that approximates $Q_\varepsilon(f)$. The subsequent work showed, among other things, that for infinitely many n, there is a function f with $\text{bm-deg}_{1/3}(f) = O(\sqrt{n})$ and $Q_{1/3}(f) = \Omega(n)$, thereby also ruling out the possibility that this degree measure validates possibility (1). The natural next question then asks if there is another refined notion of polynomial degree that approximates quantum query complexity [AAI+16, Open problem 3].

---

of $\{0,1\}$ and let $p(x) = 1 - 2q(x)$ (because p is defined as $\mathbb{E}[\mathcal{A}^{\pm}(x)] = 1 - 2\Pr[\mathcal{A}^{\pm}(x) = -1]$), then we get that $|p(x) - f^{\pm}(x)| = 2|q(x) - f(x)| \leq 2\varepsilon$.

[2]The first quantum lower bound for the search problem was proven by Bennett et al. [BBBV97] using the so-called hybrid method. Beals et al. [BBC+01] reproved their result using the polynomial method.

In the direction of the second avenue, [AAI+16] showed a surprising converse to the polynomial method for bounded quadratic polynomials. Say that a polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ is *bounded* if it satisfies $p(x) \in [-1, 1]$ for all $x \in \{-1, 1\}^n$.

4.1.1. Theorem (Aaronson et al.). *There exists an absolute constant $C \in (0, 1]$ such that the following holds. For every bounded quadratic polynomial p, there exists a one-query quantum algorithm that, on input $x \in \{-1, 1\}^n$, returns a random sign with expectation $Cp(x)$.*

This implies that possibility (2) holds true for quadratic polynomials. It also leads to the problem of finding a similar converse for higher-degree polynomials, asking for instance whether two-query quantum algorithms are equivalent to quartic polynomials [AAI+16, Open problem 1].

Organization. In Section 4.2, we give an overview of our results and briefly sketch our proof techniques. In Section 4.3, we give a brief introduction to normed vector spaces and C^*-algebras. In Section 4.4, we prove our main theorem characterizing quantum query algorithms. In Section 4.5, we explain the separation obtained for higher-degree forms. In Section 4.6, we give a short proof of the main theorem in Aaronson et al. [AAI+16]. Finally, in Section 4.7 we present some directions for future work.

4.2 Our results

In this chapter, we address the above-mentioned two problems. Our first result is a new notion of polynomial degree that gives a tight characterization of quantum query complexity (Definition 4.2.3 and Corollary 4.2.4 below), giving an answer to [AAI+16, Open problem 3]. Using this characterization, we show that there is no generalization of Theorem 4.1.1 to higher-degree polynomials, in the sense that there is no absolute constant $C \in (0, 1]$ for which the analogous statement holds true. This gives a partial answer to [AAI+16, Open problem 1], ruling out a strong kind of equivalence. Finally, we give a simplified shorter proof of Theorem 4.1.1. Below we explain our results in more detail.

4.2.1 Quantum algorithms are completely bounded forms

For the rest of the discussion, all polynomials will be assumed to be bounded, real and $(2n)$-variate if not specified otherwise. We refer to a homogeneous polynomial as a *form*. For $\alpha \in \{0, 1, 2, \ldots\}^{2n}$ and $x \in \mathbb{R}^{2n}$, we write $|\alpha| = \alpha_1 + \cdots + \alpha_{2n}$ and $x^\alpha = x_1^{\alpha_1} \cdots x_{2n}^{\alpha_{2n}}$. Then, every form p of degree t can be written as

$$p(x) = \sum_{\substack{\alpha \in \{0,1,\ldots,t\}^{2n}: \\ |\alpha| = t}} c_\alpha x^\alpha, \tag{4.1}$$

where c_α are some real coefficients. Our new notion of polynomial degree is based on a characterization of quantum query algorithms in terms of forms satisfying a certain norm constraint. The norm we assign to a form as in Eq. (4.1) is given by a norm of the symmetric t-tensor $T_p \in \mathbb{R}^{2n\times\cdots\times 2n}$ with $(i_1,\ldots,i_t)$-coordinate

$$(T_p)_{i_1,\ldots,i_t} = \frac{c_{e_{i_1}+\cdots+e_{i_t}}}{|\{i_1,\ldots,i_t\}|!}, \tag{4.2}$$

where e_i is the ith standard basis vector for $\mathbb{R}^{2n}$ and $|\{i_1,\ldots,i_t\}|$ denotes the number of distinct elements in the set $\{i_1,\ldots,i_t\}$. Note that p can then also be written as

$$p(x) = \sum_{i_1,\ldots,i_t=1}^{2n} (T_p)_{i_1,\ldots,i_t} x_{i_1}\cdots x_{i_t}. \tag{4.3}$$

The relevant norm of T_p is in turn given in terms of an infimum over decompositions of the form $T_p = \sum_{\sigma\in S_t} T^\sigma \circ \sigma$, where the sum is over permutations of $\{1,\ldots,t\}$, each T^σ is a t-tensor, and $T^\sigma\circ\sigma$ is the permuted version of T^σ given by

$$(T^\sigma\circ\sigma)_{i_1,\ldots,i_t} = T^\sigma_{i_{\sigma(1)},\ldots,i_{\sigma(t)}}.$$

Finally, the actual norm is based on the *completely bounded norm* of each of the T^σ. Given a t-tensor $T\in\mathbb{R}^{2n\times\cdots\times 2n}$, its completely bounded norm $\|T\|_{\mathrm{cb}}$ is given by the supremum over positive integers k and collections of $k\times k$ unitary matrices $U_1(i),\ldots,U_t(i)$, for $i\in[2n]$, of the operator norm

$$\Big\| \sum_{i_1,\ldots,i_t=1}^{2n} T_{i_1,\ldots,i_t} U_1(i_1)\cdots U_t(i_t)\Big\|. \tag{4.4}$$

4.2.1. Definition (Completely bounded norm of a form). Let p be a form of degree t and let T_p be the symmetric t-tensor as in Eq. (4.2). Then, the *completely bounded norm* of p is defined by

$$\|p\|_{\mathrm{cb}} = \inf\Big\{ \sum_{\sigma\in S_t} \|T^\sigma\|_{\mathrm{cb}} \;:\; T_p = \sum_{\sigma\in S_t} T^\sigma\circ\sigma \Big\}. \tag{4.5}$$

This norm was originally introduced in the general context of tensor products of operator spaces in [OP99]. In that framework, the definition considered here corresponds to a particular operator space based on ℓ_1^n, but we shall not use this fact here. Our characterization of quantum query algorithms is as follows.

4.2.2. Theorem (Characterization of quantum algorithms). *Let t be a positive integer and $\beta:\{-1,1\}^n\to[-1,1]$. Then, the following are equivalent.*

1. *There exists a form p of degree $2t$ such that $\|p\|_{\mathrm{cb}} \leq 1$ and $p((x, 1^n)) = \beta(x)$ for every $x \in \{-1,1\}^n$, where $1^n \in \mathbb{R}^n$ is the all-ones vector.*

2. *There exists a t-query quantum algorithm that, on input $x \in \{-1,1\}^n$, returns a random sign with expected value $\beta(x)$.*

It may be observed that the content of the polynomial method is contained in the above statement, since every $(2n)$-variate form p defines an n-variate polynomial given by $q(x) = p((x, 1^n))$. The above theorem refines the polynomial method in the sense that quantum algorithms can only yield polynomials of the form $q(x) = p((x, 1^n))$ where p has completely bounded norm at most one.

Our proof is based on a fundamental representation theorem of Christensen and Sinclair [CS87] concerning multilinear forms on C^*-algebras that generalizes the well-known Stinespring representation theorem for quantum channels to multilinear forms (see also [PS87] and [Pis03, Chapter 5]).

Completely bounded approximate degree. Theorem 4.2.2 motivates the following new notion of approximate degree for partial Boolean functions.

4.2.3. Definition (Completely bounded approximate degree). For every $D \subseteq \{-1,1\}^n$, let $f : D \to \{-1,1\}$ be a (possibly partial) Boolean function and let $\varepsilon \geq 0$. Then, the *ε-completely bounded approximate degree* of f, denoted $\widetilde{\deg}_{\varepsilon}(f)$, is the smallest positive integer t for which there exists a form p of degree $2t$ such that $\|p\|_{\mathrm{cb}} \leq 1$ as in Eq. (4.5) and we have $|p((x, 1^n)) - f(x)| \leq 2\varepsilon$ for every $x \in D$.

As a corollary of Theorem 4.2.2, we get the following characterization of quantum query complexity.

4.2.4. Corollary. *For every $D \subseteq \{-1,1\}^n$, $f : D \to \{-1,1\}$ and $\varepsilon \geq 0$, we have $\widetilde{\deg}_{\varepsilon}(f) = Q_{\varepsilon}(f)$.*

4.2.2 Constant query quantum algorithms

Separations for higher-degree forms. Theorem 4.1.1 follows from our Theorem 4.2.2 and the fact that for every bounded quadratic form $p(x) = x^{\mathsf{T}}Ax$, the matrix A has completely bounded norm bounded from above by an absolute constant (independent of n); this is discussed in more detail below. If the same were true for the tensors T_p corresponding to higher-degree forms p then Theorem 4.2.2 would give higher-degree extensions of Theorem 4.1.1. Unfortunately, this will turn out to be false for polynomials of degrees greater than 3. Bounded forms whose associated tensors have unbounded completely bounded norm appeared before in the work of Smith [Smi88], who gave an explicit example with

completely bounded norm $\sqrt{\log n}$. Since $\|p\|_{\mathrm{cb}}$ involves an infimum over decompositions of T_p, this does not yet imply a counterexample to higher-degree versions of Theorem 4.1.1. However, such counterexamples are implied by recent work on Bell inequalities, multiplayer XOR games in particular. It is not difficult to see that $\|p\|_{\mathrm{cb}}$ is bounded from below by the so-called *jointly completely bounded norm* of the tensor T_p, a quantity that in quantum information theory is better known as the entangled bias of the XOR game whose (unnormalized) game tensor is given by T_p. One obtains this quantity by inserting tensor products between the unitaries appearing in Eq. (4.4). Pérez-García et al. [PGWP+08] and Vidick and the second author [BV13] gave examples of bounded cubic forms with unbounded jointly completely bounded norm. Both constructions are non-explicit, the first giving a completely bounded norm of order $\Omega((\log n)^{1/4})$ and the latter of order $\widetilde{\Omega}(n^{1/4})$.

In this chapter, we explain how to get a larger separation by means of a much simpler (although still non-explicit) construction and show that a bounded cubic form p given by a suitably normalized random sign tensor has completely bounded norm $\|p\|_{\mathrm{cb}} = \Omega(\sqrt{n})$ with high probability (Theorem 4.5.1). The result presented here is not new, but it follows from the existence of commutative operator algebras which are not Q-algebras. Here, we present a self-contained proof which follows the same lines as in [DJT95, Theorem 18.16] and, in addition, we prove the result with high probability (rather than just the existence of such trilinear forms). We also explain how to obtain from this result quartic examples by embedding into 3-dimensional "tensor slices", which in turn imply counterexamples to a quartic versus two-query version of Theorem 4.1.1. Finally, we prove that the separations that we obtain are in fact optimal.

Short proof of Theorem 4.1.1. As shown in [AAI+16], Theorem 4.1.1 is yet another surprising consequence of the ubiquitous Grothendieck inequality [Gro53] (Theorem 4.5.8 below), well known for its relevance to Bell inequalities [Tsi87, CHTW04] and combinatorial optimization [AN06, KN12], not to mention its fundamental importance to Banach spaces [Pis12]. An equivalent formulation of Grothendieck's inequality again recovers Theorem 4.1.1 for quadratic forms $p(x) = x^{\mathsf{T}}Ax$ given by a matrix $A \in \mathbb{R}^{n\times n}$ satisfying a certain norm constraint $\|A\|_{\ell_\infty\to\ell_1} \leq 1$, which in particular implies that p is bounded (see Section 4.3 for more on this norm). Indeed, in that case Grothendieck's inequality implies that $\|A\|_{\mathrm{cb}} \leq K_G$ for some absolute constant $K_G \in (1,2)$ (which is independent of n and A).[3] Normalizing by K_G^{-1}, one obtains Theorem 4.1.1 with $C = K_G^{-1}$ for such quadratic forms from Theorem 4.2.2. The general version of Theorem 4.1.1 for quadratic polynomials follows from this via a so-called decoupling argument (see Section 4.6). This arguably does not simplify the original proof of Theorem 4.1.1,

[3] K_G is the Grothendieck's constant, whose precise value is unknown and is known [Ree91, BMMN13] to lie in the interval $1.6769\cdots \leq K_G < 1.7822\cdots$.

as Theorem 4.2.2 relies on deep results itself.

In Section 4.6 we give a short simplified proof, showing that Theorem 4.1.1 follows almost directly from a "factorization version" of Grothendieck's inequality (Theorem 4.6.3) that follows from the more standard version (Theorem 4.5.8). The factorization version was used in the original proof as well, but only as a lemma in a more intricate argument. In computer science, this factorization version already found applications in [Tro09, LLV15]. This appears to be its first occurrence in quantum computing.

4.2.3 Related work

Although there was no converse to the polynomial method until our work, equivalences between quantum algorithms and polynomials have been studied before in certain models of computation. For example, we do know of such characterization in the model of non-deterministic query complexity [Wol03], unbounded-error query complexity [BVW07, MNR11] and quantum query complexity in expectation [KLW15]. We remark here that in all these settings, the quantum algorithms constructed from polynomials were *non-adaptive* algorithms, i.e., the quantum algorithm begins with a quantum state, repeatedly applies the oracle some fixed number of times and then performs a projective measurement. Crucially, these algorithms do not contain interlacing unitaries that are present in the standard model of quantum query complexity, hence are known to be a much weaker class of algorithms.

Our main result is yet another demonstration of the expressive power of C^*-algebras and operator space theory in quantum information theory; for a survey on applications of these areas to two-prover one-round games, see [PV16]. The appearance of Q-algebras (mentioned in the above paragraph on separations) is also not a first in quantum information theory, see for instance [PGWP+08, BBLV12, BBLV13].

4.3 Preliminaries

Notation. For $x \in \mathbb{C}^n$, let $\mathrm{Diag}(x)$ be the $n \times n$ diagonal matrix whose diagonal is x. Given a matrix $X \in \mathbb{C}^{n \times n}$, let $\mathrm{diag}(X) \in \mathbb{C}^n$ denote its diagonal vector. For $x \in \{0,1\}^n$, denote $(-1)^x = ((-1)^{x_1}, \ldots, (-1)^{x_n})$. Let $e_1, e_2, \ldots, e_n \in \mathbb{C}^n$ be the standard basis vectors and let $E_{ij} = e_i e_j^*$. Let $1^n = (1, \ldots, 1)$ and $0^n = (0, \ldots, 0)$ denote the n-dimensional all-ones (resp. all-zeros) vector.

Normed vector spaces. For parameter $p \in [1, \infty)$, the p-norm of a vector $x \in \mathbb{R}^n$ is defined by $\|x\|_{\ell_p} = (|x_1|^p + \cdots + |x_n|^p)^{1/p}$ and for $p = \infty$ by $\|x\|_{\ell_\infty} = \max\{|x_i| : i \in [n]\}$. Denote the n-dimensional Euclidean unit ball by $B_2^n = \{x \in \mathbb{R}^n : \|x\|_{\ell_2} \leq 1\}$. For a matrix $A \in \mathbb{R}^{n \times n}$, denote the standard operator norm

by $\|A\|$ and define

$$\|A\|_{\ell_\infty \to \ell_1} = \sup\left\{\|Ax\|_{\ell_1} : \|x\|_{\ell_\infty} \leq 1\right\}.$$

By linear programming duality, observe that the right-hand side of equality above can be written as

$$\sup\left\{\|Ax\|_{\ell_1} : \|x\|_{\ell_\infty} \leq 1\right\} = \sup_{x,y \in \{-1,1\}^n} x^{\mathsf{T}} A y.$$

We denote the norm of a general normed vector space X by $\|\cdot\|_X$, if there is a danger of ambiguity. Denote by $\mathbf{1}_X$ the identity map on X and by $\mathbf{1}_d$ the identity map on $\mathbb{C}^d$. For normed vector spaces X, Y, let $L(X, Y)$ be the collection of all linear maps $T : X \to Y$. We will use the notation $L(X)$ as a shorthand for $L(X, X)$. The (operator) norm of a linear map $T \in L(X, Y)$ is given by $\|T\| = \sup\{\|T(x)\|_Y : \|x\|_X \leq 1\}$. Such a map is an *isometry* if $\|T(x)\|_Y = \|x\|_X$ for every $x \in X$ and a *contraction* if $\|T(x)\|_Y \leq \|x\|_X$ for every $x \in X$. Throughout we endow $\mathbb{C}^d$ with the standard Euclidean norm. Note that the space $L(\mathbb{C}^d)$ is naturally identified with the set of $d \times d$ matrices, sometimes denoted $M_d(\mathbb{C})$, and we use the two notations interchangeably. For a Hilbert space $\mathcal{H}$, we endow $\mathcal{H} \otimes \mathbb{C}^d$ with the norm given by the inner product $\langle f \otimes a, g \otimes b\rangle = \langle f, g\rangle_{\mathcal{H}} \langle a, b\rangle$, making this space isometric to $\mathcal{H} \oplus \cdots \oplus \mathcal{H}$ (d times). This can be extended linearly to the entire domain. Similarly, we endow $L(\mathcal{H}) \otimes L(\mathbb{C}^d)$ with the operator norm of the space $L(\mathcal{H} \otimes \mathbb{C}^d)$ of linear operators on the Hilbert space $\mathcal{H} \otimes \mathbb{C}^d$; with some abuse of notation, we shall identify the two spaces of operators.

C^*-algebras. We collect a few basic facts of C^*-algebras that we use later and refer to [Arv12] for an extensive introduction. A C^*-algebra $\mathcal{X} = (X, \cdot, *)$ is a normed complex vector space X, complete with respect to its norm (i.e., a Banach space), that is endowed with two operations in addition to the standard vector-space addition and scalar multiplication operations:

1. an associative multiplication $\cdot : X \times X \to X$, denoted $x \cdot y$ for $x, y \in X$, that is distributive with respect to the vector space addition and continuous with respect to the norm of X, which is to say that $\|x \cdot y\|_X \leq \|x\|_X \|y\|_X$ for all $x, y \in X$;

2. an involution $* : X \to X$, that is, a conjugate linear map that sends $x \in X$ to (a unique) $x^* \in X$ satisfying $(x^*)^* = x$ and $(xy)^* = y^* x^*$ for every $x, y \in X$, and such that $\|x \cdot x^*\|_X = \|x\|_X^2$.

Every finite-dimensional normed vector space is a Banach space. A C^*-algebra $\mathcal{X}$ is *unital* if it has a multiplicative identity, denoted $\mathbf{1}_{\mathcal{X}}$. The most important example of a unital C^*-algebra is $M_n(\mathbb{C})$, where the involution operator is the conjugate-transpose and the norm is the operator norm. A linear map $\pi : \mathcal{X} \to \mathcal{Y}$

from one C^*-algebra $\mathcal{X}$ to another $\mathcal{Y}$ is a *$*$-homomorphism* if it preserves the multiplication operation, $\pi(xy) = \pi(x)\pi(y)$, and satisfies $\pi(x)^* = \pi(x^*)$ for all $x, y \in \mathcal{X}$. For a complex Hilbert space $\mathcal{H}$, a mapping $\pi : \mathcal{X} \to L(\mathcal{H})$ is a *$*$-representation* if it is a $*$-homomorphism. An important fact is the Gelfand–Naimark Theorem [Mur14, Theorem 3.4.1] asserting that every C^*-algebra admits an isometric (that is, norm-preserving) $*$-representation for some complex Hilbert space. Suppose $\mathcal{X} = (X, \cdot_X, *), \mathcal{Y} = (Y, \cdot_Y, \dagger)$ are C^*-algebras, then the *tensor product* $\mathcal{X} \otimes \mathcal{Y}$ is also a C^*-algebra defined in terms of the standard tensor product of the vector spaces $X \otimes Y$ with the associative multiplication $\cdot_{XY}$ and involution operator $\diamond$ defined as: $(x \otimes y) \cdot_{XY} (x' \otimes y') = (x \cdot_X x') \otimes (y \cdot_Y y')$ and involution $(x \otimes y)^\diamond = x^* \otimes y^\dagger$. This can then be extended linearly to the entire domain.

Completely bounded norms. We also collect a few basic facts about completely bounded norms that we use later and refer to [Pau02] for an extensive introduction. For a C^*-algebra $\mathcal{X}$ and positive integer d, we denote by $M_d(\mathcal{X})$ the set of d-by-d matrices with entries in $\mathcal{X}$. Note that this set can naturally be identified with the algebraic tensor product $\mathcal{X} \otimes L(\mathbb{C}^d)$, that is, the linear span of all elements of the form $x \otimes M$, where $x \in X$ and $M \in L(\mathbb{C}^d)$. We shall endow $M_d(\mathcal{X})$ with a norm induced by an isometric $*$-representation π of $\mathcal{X}$ into $L(\mathcal{H})$ for a Hilbert space $\mathcal{H}$. The linear map $\pi \otimes \mathbf{1}_{L(\mathbb{C}^d)}$ sends elements in $M_d(\mathcal{X})$ (or $\mathcal{X} \otimes L(\mathbb{C}^d)$) to elements (operators) in $L(\mathcal{H} \otimes \mathbb{C}^d)$. The norm of an element $A \in M_d(\mathcal{X})$ is then defined to be $\|A\| = \|(\pi \otimes \mathbf{1}_{L(\mathbb{C}^d)})(A)\|$. The notation $\|A\|$ reflects the fact that this norm is in fact independent of the particular $*$-representation π. Based on this, we can define a norm on linear maps $\sigma : \mathcal{X} \to L(\mathcal{H})$ as follows:

$$\|\sigma\|_{\mathrm{cb}} = \sup \left\{ \frac{\|(\sigma \otimes \mathbf{1}_{L(\mathbb{C}^d)})(A)\|}{\|A\|} : d \in \mathbb{N},\ A \in \mathcal{X} \otimes L(\mathbb{C}^d),\ A \neq 0 \right\}$$

Tensors and multilinear forms. For vector spaces X, Y over the same field and positive integer t, recall that a mapping

$$T : \underbrace{X \times \cdots \times X}_{t \text{ times}} \to Y$$

is *t-linear* if for every $x_1, \ldots, x_t \in X$ and $i \in [t]$, the map

$$y \mapsto T(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_t)$$

is linear. A *t-tensor* of dimension n is a map $T : [n] \times \cdots \times [n] \to \mathbb{C}$, which can alternatively be identified by n^t complex numbers $T = (T_{i_1,\ldots,i_t})_{i_1,\ldots,i_t=1}^n \in \mathbb{C}^{n \times \cdots \times n}$.

With abuse of notation we also identify a t-tensor $T \in \mathbb{C}^{n\times\cdots\times n}$ with the t-linear form $T : \mathbb{C}^n \times \cdots \times \mathbb{C}^n \to \mathbb{C}$ given by

$$T(x_1, \ldots, x_t) = \sum_{i_1,\ldots,i_t=1}^{n} T_{i_1,\ldots,i_t} x_1(i_1) \cdots x_t(i_t).$$

Define the norm of a t-tensor $T \in \mathbb{R}^{n\times\cdots\times n}$ by

$$\|T\|_{\ell_\infty,\ldots,\ell_\infty} = \sup\Big\{\Big|\sum_{i_1,\ldots,i_t=1}^{n} T_{i_1,\ldots,i_t} x_1(i_1)\cdots x_t(i_t)\Big| : x_1, \ldots, x_t \in \{-1,1\}^n\Big\}.$$

Next, we introduce the *completely bounded norm* of a t-linear form $T : \mathcal{X} \times \cdots \times \mathcal{X} \to \mathbb{C}$ on a C^*-algebra $\mathcal{X}$. First, we use the standard identification of such forms with the linear form on the tensor product $\mathcal{X} \otimes \cdots \otimes \mathcal{X}$ given by $T(x_1 \otimes \cdots \otimes x_t) = T(x_1, \ldots, x_t)$. We consider a bilinear map $\odot : \big(\mathcal{X} \otimes L(\mathbb{C}^d), \mathcal{X} \otimes L(\mathbb{C}^d)\big) \to \mathcal{X} \otimes \mathcal{X} \otimes L(\mathbb{C}^d)$ for every positive integer d defined as follows. For $x, y \in \mathcal{X}$ and $M_x, M_y \in L(\mathbb{C}^d)$, let

$$(x \otimes M_x) \odot (y \otimes M_y) = (x \otimes y) \otimes (M_x M_y).$$

Observe that this operation changes the order of the tensor factors and *multiplies* M_x with M_y. This operation is associative but *not* commutative. Extend the definition of the $\odot$ operation bi-linearly to its entire domain. Define the t-linear map $T_d : M_d(\mathcal{X}) \times \cdots \times M_d(\mathcal{X}) \to L(\mathbb{C}^d)$ by

$$T_d(A_1 \ldots, A_t) = \big(T \otimes \mathbf{1}_{L(\mathbb{C}^d)}\big)(A_1 \odot \cdots \odot A_t).$$

The completely bounded norm of T is now defined by

$$\|T\|_{\mathrm{cb}} = \sup\Big\{\big\|T_d(A_1, \ldots, A_t)\big\| : d \in \mathbb{N},\ \ A_j \in M_d(\mathcal{X}),\ \ \|A_j\| \le 1\Big\}.$$

Note that the definition given in Eq. (4.4) corresponds to the particular case where the C^*-algebra $\mathcal{X}$ is formed by the $n \times n$ diagonal matrices. Since every square matrix with operator norm at most 1 is a convex combination of unitary matrices (by the Russo-Dye Theorem)[4], the completely bounded norm can also be defined by taking the supremum over unitaries $A_j \in M_d(\mathcal{X})$. The completely bounded norm can be defined more generally for multilinear maps into $L(\mathcal{H})$, for some Hilbert space $\mathcal{H}$, to yield the definition of this norm for linear maps given above, but we will not use this here.

[4] A precise statement and short proof of the Russo-Dye theorem can be found in [Gar84].

Quantum query complexity. We have discussed the quantum query model in detail in Chapter 2. Here, we make a remark about the phase oracles used there (in particular Section 2.3.2). The unitary transformation of the phase oracle $O_{x,\pm}$ corresponds to $O_{x,\pm} : |b, i\rangle \to (-1)^{b \cdot x_i} |b, i\rangle$ (applied to register (A, Q) in Fig. 2.5). In the remaining part of this chapter, we abuse notation and let $O_{x,\pm}$ correspond to the (controlled) unitary $\mathrm{Diag}(((-1)^x, 1^n))$ (instead of $\mathrm{Diag}((1^n, (-1)^x))$). Additionally, to avoid having to write $(-1)^x$ later on, we shall work in the equivalent setting where Boolean functions send $\{-1, 1\}^n$ to $\{-1, 1\}$.

4.4 Characterizing quantum query algorithms

In this section we prove Theorem 4.2.2. The main ingredient of the proof is the following celebrated representation theorem by Christensen and Sinclair [CS87] showing that completely-boundedness of a multilinear form is equivalent to the existence of an exceedingly nice factorization.

4.4.1. Theorem (Christensen–Sinclair). *Let t be a positive integer and let $\mathcal{X}$ be a C^*-algebra. Then, for every t-linear form $T : \mathcal{X} \times \cdots \times \mathcal{X} \to \mathbb{C}$, we have $\|T\|_{\mathrm{cb}} \leq 1$ if and only if there exist Hilbert spaces $\mathcal{H}_0, \ldots, \mathcal{H}_{t+1}$ where $\mathcal{H}_0 = \mathcal{H}_{t+1} = \mathbb{C}$, $*$-representations $\pi_i : \mathcal{X} \to L(\mathcal{H}_i)$ for each $i \in [t]$ and contractions $V_i \in L(\mathcal{H}_i, \mathcal{H}_{i-1})$, for each $i \in [t+1]$ such that for every $x_1, \ldots, x_t \in \mathcal{X}$, we have*

$$T(x_1, \ldots, x_t) = V_1 \pi_1(x_1) V_2 \pi_2(x_2) V_3 \cdots V_t \pi_t(x_t) V_{t+1}. \tag{4.6}$$

We first show how the above result simplifies when restricting to the special case in which the C^*-algebra $\mathcal{X}$ is formed by the set of diagonal n-by-n matrices.

4.4.2. Corollary. *Let m, n, t be positive integers such that $t \geq 2$ and $m = n^t$. Let $T \in \mathbb{C}^{n \times \cdots \times n}$ be a t-tensor. Then $\|T\|_{\mathrm{cb}} \leq 1$ if and only if there exist a positive integer d, unit vectors $u, v \in \mathbb{C}^m$ and contractions $U_i, V_i \in L(\mathbb{C}^m, \mathbb{C}^{dn})$ such that for every $x_1, \ldots, x_t \in \mathbb{C}^n$, we have*

$$T(x_1, \ldots, x_t) = u^* U_1^* \big(\mathrm{Diag}(x_1) \otimes \mathbf{1}_d \big) V_1 \cdots U_t^* \big(\mathrm{Diag}(x_t) \otimes \mathbf{1}_d \big) V_t v. \tag{4.7}$$

The proof of the above corollary uses the following fact about the completely bounded norm of $*$-representations of C^*-algebras [Pis03, Theorem 1.6].

4.4.3. Lemma. *Let $\mathcal{X}$ be a finite-dimensional C^*-algebra, $\mathcal{H}, \mathcal{H}'$ be Hilbert spaces, $\pi : \mathcal{X} \to L(\mathcal{H})$ be a $*$-representation and $U \in L(\mathcal{H}, \mathcal{H}')$ and $V \in L(\mathcal{H}', \mathcal{H})$ be linear maps. Then the map $\sigma : \mathcal{X} \to L(\mathcal{H}')$, defined as $\sigma(x) = U \pi(x) V$, satisfies that $\|\sigma\|_{\mathrm{cb}} \leq \|U\| \|V\|$.*

In addition, we use the famous Fundamental Factorization Theorem [Pau02, Theorem 8.4]. Below we state the theorem when restricted to finite-dimensional spaces (see also the remark after [JKP09, Theorem 16]).

4.4.4. THEOREM (Fundamental factorization theorem). *Let $\sigma : L(\mathbb{C}^n) \to L(\mathbb{C}^m)$ be a linear map and let $d = nm$. Then there exist $U, V \in L(\mathbb{C}^m, \mathbb{C}^{dn})$ such that $\|U\|\|V\| \leq \|\sigma\|_{\mathrm{cb}}$ and for every $M \in L(\mathbb{C}^n)$, we have $\sigma(M) = U^*(M \otimes \mathbf{1}_d)V$.*

Proof of Corollary 4.4.2. The set $\mathcal{X} = \mathrm{Diag}(\mathbb{C}^n)$ of diagonal matrices is a (finite-dimensional) C^*-algebra (endowed with the standard matrix product and conjugate-transpose involution). Define the t-linear form $R : \mathcal{X} \times \cdots \times \mathcal{X} \to \mathbb{C}$ by $R(X_1, \ldots, X_t) = T(\mathrm{diag}(X_1), \ldots, \mathrm{diag}(X_t))$.

We first show that $\|R\|_{\mathrm{cb}} = \|T\|_{\mathrm{cb}}$. Observe that for every positive integer d, the set $\{B \in M_d(\mathcal{X}) : \|B\| \leq 1\}$ can be identified with the set of block-diagonal matrices $B = \sum_{i=1}^n E_{i,i} \otimes B(i)$ of size $nd \times nd$ and blocks $B(1), \ldots, B(n)$ of size $d \times d$ satisfying $\|B(i)\| \leq 1$ for all $i \in [n]$. It then follows that

$$\begin{aligned} R_d(B_1, \ldots, B_t) &= \sum_{i_1, \ldots, i_t = 1}^{n} R(E_{i_1,i_1}, \ldots, E_{i_t,i_t}) B_1(i_1) \cdots B_t(i_t) \\ &= \sum_{i_1, \ldots, i_t = 1}^{n} T_{i_1, \ldots, i_t} B_1(i_1) \cdots B_t(i_t), \end{aligned}$$

which shows that $\|R\|_{\mathrm{cb}} = \|T\|_{\mathrm{cb}}$.

Next, we show that Eq. (4.6) is equivalent to Eq. (4.7). The fact that Eq. (4.7) implies Eq. (4.6) follows immediately from the fact that the map $\mathrm{Diag}(x) \mapsto \mathrm{Diag}(x) \otimes \mathbf{1}_d$ is clearly a $*$-representation for $\mathcal{X} = \mathrm{Diag}(\mathbb{C}^n)$. We now show that by fixing $\mathcal{X} = \mathrm{Diag}(\mathbb{C}^n)$, there exists Hilbert spaces $\mathcal{H}_i$, $*$-representations π_i and contractions V_i such that Eq. (4.6) equals Eq. (4.7). Assume Eq. (4.6). Without loss of generality, we may assume that each of the Hilbert spaces $\mathcal{H}_1, \ldots, \mathcal{H}_t$ has dimension at least m. If not, we can expand the dimensions of the ranges and domains of the representations π_i and contractions V_i by dilating with appropriate isometries into larger Hilbert spaces ("padding with zeros"). For each $i \in [t]$, let $S_i \subseteq \mathcal{H}_i$ be the subspace

$$S_i = \mathrm{Span}\left\{\pi_i(x_i) V_{i+1} \cdots V_t \pi_t(x_t) V_{t+1} : x_i, \ldots, x_t \in \mathcal{X}\right\}.$$

Since $\dim(\mathcal{X}) = n$, we have that $\dim(S_i) \leq m$ (since $V_i \in L(\mathbb{C}^m, \mathbb{C}^{dn})$). For each $i \in [t]$, let $Q_i \in L(\mathbb{C}^m, \mathcal{H}_i)$ be an isometry such that $S_i \subseteq \mathrm{Im}(Q_i)$. Note that V_{i+1} is a vector in the unit ball of $\mathcal{H}_t$. Let $Q_{t+1} \in L(\mathbb{C}^m, \mathcal{H}_t)$ be an isometry such that $\mathrm{Im}(V_{t+1}) \subseteq \mathrm{Im}(Q_{t+1})$. Note that for each $i \in [t+1]$, the map $Q_i Q_i^*$ acts as the identity on $\mathrm{Im}(Q_i)$. For each $i \in \{2, \ldots, t\}$ define the map $\sigma_i : \mathcal{X} \to L(\mathbb{C}^m)$ by $\sigma_i(x) = Q_i^* V_i \pi_i(x) Q_{i+1}$ and $\sigma_1(x) = Q_1^* \pi_1(x) Q_2$. Finally define $u = Q_1^* V_1^*$ and $v = Q_{t+1}^* V_{t+1}$. Then the right-hand side of Eq. (4.6) can be written as

$$u^* \sigma_1(x_1) \cdots \sigma_t(x_t) v.$$

It follows from Lemma 4.4.3 that $\|\sigma_i\|_{\mathrm{cb}} \leq 1$. Let $\sigma_i' : L(\mathbb{C}^n) \to L(\mathbb{C}^m)$ be the linear map given by $\sigma_i'(M) = \sigma_i(\mathrm{Diag}(M_{11}, \ldots, M_{nn}))$ for every $M \in L(\mathbb{C}^m)$. Then,

for every diagonal matrix $x \in \mathcal{X}$, we have $\sigma_i(x) = \sigma'_i(x)$ and $\|\sigma'_i\|_{\mathrm{cb}} = \|\sigma_i\|_{\mathrm{cb}}$. It follows from Theorem 4.4.4 that there exist a positive integer d_i and contractions $U_i, V_i : L(\mathbb{C}^m, \mathbb{C}^{dn})$ such that $\sigma_i(x) = U_i^*(x \otimes \mathbf{1}_{d_i})V_i$ for every $x \in \mathcal{X}$. We can take all d_i equal to $d = \max_i\{d_i\}$ by suitably dilating the contractions U_i, V_i. Setting $u' = u/\|u\|_2$ and $U'_1 = \|u\|_2 U_1$, and similarly defining v', V'_{i+1} shows that Eq. (4.6) implies Eq. (4.7). □

Corollary 4.4.2 implies the following lemma, from which Theorem 4.2.2 easily follows.

4.4.5. Lemma. *Let $\beta : \{-1,1\}^n \to [-1,1]$ and t be a positive integer. Then the following are equivalent.*

1. *There exists a $(2t)$-tensor $T \in \mathbb{R}^{2n \times \cdots \times 2n}$ such that $\|T\|_{\mathrm{cb}} \leq 1$ and for every $x \in \{-1,1\}^n$ and $y = (x, 1^n)$, we have*

$$\sum_{i_1,\ldots,i_{2t}=1}^{2n} T_{i_1,\ldots,i_{2t}} y_{i_1} \cdots y_{i_{2t}} = \beta(x).$$

2. *There exists a t-query quantum algorithm that, on input $x \in \{-1,1\}^n$, returns a random sign with expected value $\beta(x)$.*

Proof. We first prove that (2) implies (1). As discussed in Section 4.3, a t-query quantum algorithm with phase oracles initializes the joint register $(\mathsf{A}, \mathsf{Q}, \mathsf{W})$ in the all-zero state, on which it then performs a sequence of unitaries $U_1, \ldots, U_t$ interlaced with queries $D(x) = \mathrm{Diag}((x, 1^n)) \otimes \mathbf{1}_{\mathsf{W}}$. Let $\{P_0, P_1\}$ be the two-outcome measurement done at the end of the algorithm and assume that it returns $+1$ on measurement outcome zero and -1 otherwise. Note that $P_0 - P_1$ is a contraction since P_0, P_1 are positive semi-definite and satisfy $P_0 + P_1 = \mathbf{1}$.

The final state of the quantum algorithm (before the measurement of register A) is

$$|\psi_x\rangle = U_t D(x) \cdots U_2 D(x) U_1 |0^m\rangle,$$

where m is the total number of qubits in the joint register $(\mathsf{A}, \mathsf{Q}, \mathsf{W})$. Hence the expected value of the measurement outcome is then given by

$$\langle\psi_x|(P_0 - P_1)|\psi_x\rangle. \tag{4.8}$$

By assumption, this expected value equals $\beta(x)$ for every $x \in \{-1,1\}^n$. For $z \in \mathbb{C}^{2n}$, denote $D'(z) = \mathrm{Diag}((z_{n+1}, \ldots, z_{2n}, z_1, \ldots, z_n)) \otimes \mathbf{1}_{\mathsf{W}}$ and $\widetilde{U}_t = U_t^*(P_0 - P_1)U_t$. Define the $(2t)$-linear form T by

$$T(y_1, \ldots, y_{2t}) = \langle 0^m | U_1^* D'(y_1) U_2^* \cdots D'(y_t) \widetilde{U}_t D'(y_{t+1}) \cdots U_2 D'(y_{2t}) U_1 | 0^m \rangle.$$

Clearly $T((x,1^n),\dots,(x,1^n)) = \beta(x)$ for every $x \in \{-1,1\}^n$. Moreover, by definition, T admits a factorization as in Eq. (4.7). It thus follows from Corollary 4.4.2 that $\|T\|_{\mathrm{cb}} \leq 1$. We turn T into a real tensor by taking its real part $T' = (T + \overline{T})/2$, where $\overline{T}$ is the coordinate-wise complex conjugate of T. Since for every $x \in \{-1,1\}^n$ and $y = (x,1^n)$, the value $T(y,\dots,y)$ is real, we have $T'(y,\dots,y) = \beta(x)$. We need to show that $\|T'\|_{\mathrm{cb}} \leq 1$. To this end, consider an arbitrary positive integer d and sequences of unitary matrices $V_1(i),\dots,V_{2t}(i)$ for $i \in [n]$, then

$$\Big\| \sum_{i_1,\dots,i_{2t}=1}^{2n} \overline{T_{i_1,\dots,i_{2t}}} V_1(i_1)\cdots V_{2t}(i_{2t}) \Big\| = \Big| \sum_{i_1,\dots,i_{2t}=1}^{2n} \overline{T_{i_1,\dots,i_{2t}}} v^* V_1(i_1)\cdots V_{2t}(i_{2t})\, w \Big|,$$

where we assumed that the unit vectors $v, w \in \mathbb{C}^d$ maximize the operator norm. Note that $\|\overline{T}\|_{\mathrm{cb}}$ is given by the supremum over d and $V_j(i)$. Taking the complex conjugate of the above summands on the right-hand side allows us to express the above absolute value as

$$\Big| \sum_{i_1,\dots,i_{2t}=1}^{2n} T_{i_1,\dots,i_{2t}} \bar{v}^* \overline{V_1(i_1)} \cdots \overline{V_{2t}(i_{2t})}\, \bar{w} \Big|, \tag{4.9}$$

where $\bar{v}, \bar{w}, \overline{V_j(i)}$ denote the coordinate-wise complex conjugates. Since each $\overline{V_j(i)}$ is still unitary, it follows that Eq. (4.9) is at most $\|T\|_{\mathrm{cb}}$ and so $\|\overline{T}\|_{\mathrm{cb}} \leq \|T\|_{\mathrm{cb}} \leq 1$. Hence, by the triangle inequality, $\|T'\|_{\mathrm{cb}} \leq (\|T\|_{\mathrm{cb}} + \|\overline{T}\|_{\mathrm{cb}})/2 \leq 1$ as desired.

Next, we show that (1) implies (2). Let T be a $(2t)$-tensor as in item (1). Then it follows from Corollary 4.4.2 that T admits a factorization as in Eq. (4.7),

$$T(y_1,\dots,y_{2t}) = u^* U_1^* \big(\operatorname{Diag}(y_1) \otimes \mathbf{1}_d\big) V_1 \cdots U_{2t}^* \big(\operatorname{Diag}(y_{2t}) \otimes \mathbf{1}_d\big) V_{2t} v. \tag{4.10}$$

Let $V_0, U_{2t+1} \in L(\mathbb{C}^m, \mathbb{C}^{2dn})$ be isometries. For each $i \in [2t+1]$, define the map $Z_i \in L(\mathbb{C}^{2dn})$ by $Z_i = V_{i-1} U_i^*$. Observe that each Z_i is a contraction and recall that unitaries are contractions. For the moment, assume for simplicity that each Z_i is in fact unitary. Define two vectors $\widetilde{u} = V_0 u$ and $\widetilde{v} = U_{2t+1} v$ and observe that these are unit vectors in $\mathbb{C}^{2dn}$. The right-hand side of Eq. (4.10) then gives us

$$\begin{aligned} &T(y_1,\dots,y_{2t}) = \\ &\widetilde{u}^* Z_1 \big(\operatorname{Diag}(y_1) \otimes \mathbf{1}_d\big) Z_2 \big(\operatorname{Diag}(y_2) \otimes \mathbf{1}_d\big) Z_3 \cdots Z_{2t} \big(\operatorname{Diag}(y_{2t}) \otimes \mathbf{1}_d\big) Z_{2t+1} \widetilde{v}. \end{aligned} \tag{4.11}$$

In particular, if we define two unit vectors

$$\begin{aligned} V_1 &= (\operatorname{Diag}(y) \otimes \mathbf{1}_d) Z_t \cdots W_2 (\operatorname{Diag}(y) \otimes \mathbf{1}_d) Z_1 \widetilde{u}, \\ V_2 &= Z_{t+1}^* (\operatorname{Diag}(y) \otimes \mathbf{1}_d) Z_{t+2}^* \cdots Z_{2t}^* (\operatorname{Diag}(y) \otimes \mathbf{1}_d) Z_{2t+1}^* \widetilde{v}, \end{aligned}$$

then $T(y,\dots,y) = |\langle V_1^*, V_2 \rangle|$. Based on this, we obtain the quantum query algorithm that prepares V_1 and V_2 in parallel, each using at most t queries. This is described in Figure 4.1.

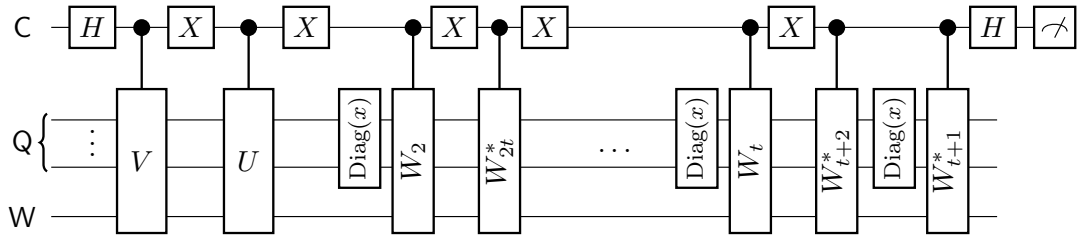


Figure 4.1: The registers $\mathsf{A}, \mathsf{Q}, \mathsf{W}$ denote the control, query and workspace registers. Let U, V be unitaries with $Z_1\widetilde{u}$ and $Z_{2t+1}^*\widetilde{v}$ as their first columns, respectively and for $x \in \{-1,1\}^n$ and $y = (x, 1^n)$, let $\mathrm{Diag}(y)$ be the query operator. The algorithm begins by initializing the joint register $(\mathsf{A}, \mathsf{Q}, \mathsf{W})$ in the all-zero state and proceeds by performing the displayed operations. The algorithm returns $+1$ if the outcome of the measurement on A equals zero and -1 otherwise.

To see why this algorithm satisfies the requirements, first note that the algorithm makes t queries to the input x. For the correctness of the algorithm, we begin by observing that before the application of the first query, the state of the joint register $(\mathsf{A}, \mathsf{Q}, \mathsf{W})$ is

$$\tfrac{1}{\sqrt{2}}(|0\rangle \otimes U|0^n\rangle + |1\rangle \otimes V|0^n\rangle) = \tfrac{1}{\sqrt{2}}(|0\rangle \otimes Z_1\widetilde{u} + |1\rangle \otimes Z_{2t+1}^*\widetilde{v}).$$

Before the final Hadamard gate, the state of the joint register is given by

$$\begin{aligned}&\frac{1}{\sqrt{2}}|0\rangle \otimes \big((\mathrm{Diag}(y) \otimes \mathbf{1}_d)Z_t \cdots W_2(\mathrm{Diag}(y) \otimes \mathbf{1}_d)Z_1\widetilde{u}\big)\\ +&\frac{1}{\sqrt{2}}|1\rangle \otimes \big(Z_{t+1}^*(\mathrm{Diag}(y) \otimes \mathbf{1}_d)Z_{t+2}^* \cdots Z_{2t}^*(\mathrm{Diag}(y) \otimes \mathbf{1}_d)Z_{2t+1}^*\widetilde{v}\big).\end{aligned}$$

A standard calculation shows that after the final Hadamard gate, the expected value of the final measurement outcome is

$$\widetilde{u}^* Z_1\big(\,\mathrm{Diag}(y) \otimes \mathbf{1}_d\big) Z_2\big(\,\mathrm{Diag}(y) \otimes \mathbf{1}_d\big) Z_3 \cdots Z_{2t}\big(\,\mathrm{Diag}(y) \otimes \mathbf{1}_d\big) Z_{2t+1}\widetilde{v}.$$

Using Eq. (4.11), it follows that the expected output of the algorithm is precisely $T((x,1^n),\ldots,(x,1^n)) = \beta(x)$.

In the general case where the Z_is are not necessarily unitary but have operator norm at most 1, we can use the linear algebra fact that there exists a unitary matrix $Z_i' \in \mathbb{C}^{4dn \times 4dn}$ that has Z_i as its upper-left corner (see [AAI+16, Lemma 7]), through which the algorithm could implement Z_i by working on a larger quantum register. Correspondingly, one could increase the dimension of $\widetilde{u}, \widetilde{v}$ by fixing their entries to be 0 in the additional dimensions. □

Using Lemma 4.4.5, we now prove our main Theorem 4.2.2.

Proof of Theorem 4.2.2. We first show that (2) implies (1). Using the equivalence in Lemma 4.4.5 (in particular (2) $\implies$ (1) in Lemma 4.4.5), there exists a $(2t)$-tensor $T \in \mathbb{R}^{2n\times\cdots\times 2n}$ such that $\|T\|_{\mathrm{cb}} \leq 1$ and for every $x \in \{-1,1\}^n$ and $y = (x, 1^n)$, we have

$$\sum_{i_1,\ldots,i_{2t}=1}^{2n} T_{i_1,\ldots,i_{2t}} y_{i_1} \cdots y_{i_{2t}} = \beta(x).$$

Define the symmetric $2t$-tensor $T' = \frac{1}{(2t)!}\sum_{\sigma\in S_{2t}} T \circ \sigma$. Let $p \in \mathbb{R}[x_1,\ldots,x_{2n}]$ be the form of degree $2t$ associated with T'. Since there is a unique symmetric tensor associated with a polynomial, it follows that $T' = T_p$ (where T_p is defined by Eq. (4.2)). Then, $p((x,1^n)) = \beta(x)$ for every $x \in \{-1,1\}^n$. Moreover, if we set $T^\sigma = T$ for each $\sigma \in S_{2t}$, it follows from the above decomposition of T_p and Definition 4.2.1 that $\|p\|_{\mathrm{cb}} \leq \|T\|_{\mathrm{cb}} \leq 1$.

Next, we show that (1) implies (2). Let p be a degree-$(2t)$ form satisfying $\|p\|_{\mathrm{cb}} \leq 1$. Suppose T_p as defined in Eq. (4.2) can be written as $T_p = \sum_{\sigma\in S_{2t}} T^\sigma \circ \sigma$ and $\sum_{\sigma\in S_{2t}} \|T^\sigma\|_{\mathrm{cb}} = \|p\|_{\mathrm{cb}} \leq 1$. Define $T = \sum_{\sigma\in S_{2t}} T^\sigma$. Then, using the triangle inequality, it follows that $\|T\|_{\mathrm{cb}} \leq \sum_{\sigma\in S_{2t}} \|T^\sigma\|_{\mathrm{cb}} \leq 1$. Also note that for every $y \in \mathbb{R}^{2n}$,

$$T(y,\ldots,y) = \sum_{\sigma\in S_{2t}} T^\sigma(y,\ldots,y) = \sum_{\sigma\in S_{2t}} (T^\sigma\circ\sigma)(y,\ldots,y) = T_p(y,\ldots,y) = p(y).$$

Using Lemma 4.4.5 (in particular (1) $\implies$ (2) in Lemma 4.4.5) for the tensor T, the theorem follows. □

We now prove Corollary 4.2.4, which is an immediate consequence of our main theorem.

Proof of Corollary 4.2.4. We first prove that $\text{cb-deg}_\varepsilon(f) \geq Q_\varepsilon(f)$. Suppose $\text{cb-deg}_\varepsilon(f) = d$, then there exists a degree-$(2d)$ form p satisfying: $|p(x,1^n) - f(x)| \leq 2\varepsilon$ for every $x \in D$ and $\|p\|_{\mathrm{cb}} \leq 1$. Let $\beta(x) = p(x,1^n)$ for every $x \in \{-1,1\}^n$. Using our characterization in Theorem 4.2.2, it follows that there exists a d-query quantum algorithm $\mathcal{A}$, that on input $x \in D$, returns a random sign with expected value $\beta(x) = p(x,1^n)$. So, our ε-error quantum algorithm for f simply runs $\mathcal{A}$ and outputs the random sign.

We next show $\text{cb-deg}_\varepsilon(f) \leq Q_\varepsilon(f)$. Suppose $Q_\varepsilon(f) = t$. Then, there exists a t-query quantum algorithm that, on input $x \in D$, outputs a random sign with expected value $\beta(x)$ satisfying $|\beta(x) - f(x)| \leq 2\varepsilon$. Note that we could also run the quantum algorithm for $x \notin D$ and let $\beta(x) \in [-1,1]$ be the expected value of the quantum algorithm for such xs. Using Theorem 4.2.2, we know that there exists a degree-$(2t)$ form p satisfying $\beta(x) = p(x,1^n)$ for every $x \in \{-1,1\}^n$ and $\|p\|_{\mathrm{cb}} \leq 1$. Clearly p satisfies satisfies the conditions of Definition 4.2.3, hence $\text{cb-deg}_\varepsilon(f) \leq t$. □

4.5 Separations for quartic polynomials

In this section we show the existence of a quartic polynomial p that is bounded but for which every two-query quantum algorithm $\mathcal{A}$ satisfying $\mathbb{E}[\mathcal{A}(x)] = Cp(x)$ for every $x \in \{-1,1\}^n$, must necessarily have $C = O(n^{-1/2})$. We show this using a (random) *cubic* form that is bounded, but whose completely bounded norm is poly(n), following a construction of [DJT95, Theorem 18.16]. This shows that Theorem 4.1.1 cannot be generalized from one-query to two-query quantum algorithms.

Given a form $p : \mathbb{R}^n \to \mathbb{R}$, we define its norm as

$$\|p\| = \sup\{|p(x)| : x \in \{-1,1\}^n\}.$$

Note that the condition $\|p\| \leq 1$ is equivalent to p being bounded.

4.5.1 Probabilistic counterexample

4.5.1. Theorem. *There exist absolute constants $C, \kappa \in (0,\infty)$ such that the following holds. Let*[5]

$$p(x) = \sum_{\alpha \in \{0,1,2,3\}^n : |\alpha|=3} c_\alpha x^\alpha$$

be a random cubic form where the coefficients c_α are independent uniformly distributed $\{-1,1\}$-valued random variables. Then, with probability $\geq 1 - Cne^{-\kappa n}$, we have $\|p\|_{\mathrm{cb}} \geq \kappa\sqrt{n}\|p\|$.

We shall use the following standard concentration-of-measure results. The first is the Hoeffding bound [Pol12, Corollary 3 (Appendix B)].

4.5.2. Lemma (Hoeffding bound). *Let $X_1, \ldots, X_m$ be independent uniformly distributed $\{-1,1\}$-random variables and let $a \in \mathbb{R}^m$. Then, for every $\tau > 0$, we have*

$$\Pr\Big[\Big|\sum_{i=1}^m a_i X_i\Big| > \tau\Big] \leq 2e^{-\frac{\tau^2}{2(a_1^2+\cdots+a_m^2)}}$$

The second result is one from random matrix theory concerning upper tail estimates for Wigner ensembles (see [Tao12, Corollary 2.3.6]).

4.5.3. Lemma. *There exist absolute constants $C, \kappa \in (0,\infty)$ such that the following holds. Let n be a positive integer and let M be a random $n \times n$ symmetric random matrix such that for $j \geq i$, the entries M_{ij} are independent random variables with mean zero and absolute value at most 1. Then, for every $\tau \geq C$, we have*

$$\Pr\big[\|M\| > \tau\sqrt{n}\big] \leq Ce^{-\kappa\tau n}.$$

[5] Recall that $|\alpha|$ in the definition of p is defined as $|\alpha| = \sum_i \alpha_i$.

We also use the following proposition.

4.5.4. PROPOSITION. *Let m, n, t be positive integers, let $p \in \mathbb{R}[x_1, \ldots, x_n]$ be a t-linear form, let $T_p \in \mathbb{R}^{n \times \cdots \times n}$ be as in Eq. (4.2) and let $A_1, \ldots, A_n \in L(\mathbb{R}^m)$ be pairwise commuting contractions. Then,*

$$\|p\|_{\mathrm{cb}} \geq \Big\| \sum_{i_1,\ldots,i_t=1}^{n} (T_p)_{i_1,\ldots,i_t} A_{i_1} \cdots A_{i_t} \Big\|.$$

Proof. Consider an arbitrary decomposition $T_p = \sum_{\sigma \in S_t} T^\sigma \circ \sigma$. Then, the definition of the completely bounded norm and triangle inequality shows that for every sequence of commuting contractions $A_1, \ldots, A_n \in L(\mathbb{R}^m)$, we have

$$\begin{aligned} \sum_{\sigma \in S_t} \|T^\sigma\|_{\mathrm{cb}} &\geq \sum_{\sigma \in S_t} \Big\| \sum_{i_1,\ldots,i_t=1}^{n} T^\sigma_{i_1,\ldots,i_t} A_{i_1} \cdots A_{i_t} \Big\| \\ &\geq \Big\| \sum_{\sigma \in S_t} \sum_{i_1,\ldots,i_t=1}^{n} T^\sigma_{i_1,\ldots,i_t} A_{i_1} \cdots A_{i_t} \Big\|. \end{aligned}$$

Since the A_i commute, the above reduces to

$$\begin{aligned} \Big\| \sum_{\sigma \in S_t} \sum_{i_1,\ldots,i_t=1}^{n} T^\sigma_{i_1,\ldots,i_t} A_{\sigma^{-1}(i_1)} \cdots A_{\sigma^{-1}(i_t)} \Big\| &= \Big\| \sum_{\sigma \in S_t} \sum_{i_1,\ldots,i_t=1}^{n} (T^\sigma \circ \sigma)_{i_1,\ldots,i_t} A_{i_1} \cdots A_{i_t} \Big\| \\ &= \Big\| \sum_{i_1,\ldots,i_t=1}^{n} (T_p)_{i_1,\ldots,i_t} A_{i_1} \cdots A_{i_t} \Big\|. \end{aligned}$$

The proposition now follows from the definition of $\|p\|_{\mathrm{cb}}$ and the fact that the decomposition of T_p was arbitrary. □

Proof of Theorem 4.5.1. We begin by showing that with probability at least $1 - 2e^{-n}$, we have $\|p\| = \max_{x \in \{-1,1\}^n} |p(x)| \leq O(n^2)$. To this end, let us fix an arbitrary $x \in \{-1, 1\}^n$. Then, $p(x)$ is a sum of at most n^3 independent uniformly distributed random $\{-1, 1\}$-random variables. It therefore follows from Lemma 4.5.2 that

$$\Pr\big[|p(x)| > 2n^2\big] \leq 2e^{-2n},$$

By the union bound over $x \in \{-1, 1\}^n$, it follows that $\|p\| > 2n^2$ with probability at most $2e^{-n}$, which gives the claim.

We now lower bound $\|p\|_{\mathrm{cb}}$. Let $\tau > 0$ be a parameter to be set later. Let $T \in \mathbb{R}^{n \times n \times n}$ be the random symmetric 3-tensor associated with p as in Eq. (4.2). For every $i \in [n]$, we define the linear map $A_i : \mathbb{R}^{2n+2} \to \mathbb{R}^{2n+2}$ by

$$\begin{cases} A_i e_0 = e_i \\ A_i e_j = \frac{1}{\tau\sqrt{n}} \sum_{k=1}^{n} T_{i,j,k} e_{k+n} \\ A_i e_{j+n} = \delta_{i,j} e_{2n+1} \\ A_i e_{2n+1} = 0. \end{cases}$$

Observe that for every $i, j, k \in [n]$, we have

$$\begin{aligned} e_{2n+1}^* A_i A_j A_k e_0 = e_{2n+1}^* A_i A_j e_k &= e_{2n+1}^* A_i \Big(\frac{1}{\tau\sqrt{n}} \sum_{k'=1}^{n} T_{j,k,k'} e_{k'+n} \Big) \\ &= e_{2n+1}^* \frac{1}{\tau\sqrt{n}} \sum_{k'=1}^{n} T_{j,k,k'} \delta_{i,k'} e_{2n+1} \\ &= \frac{1}{\tau\sqrt{n}} T_{i,j,k}. \end{aligned} \tag{4.12}$$

Since T is symmetric, it follows easily that these maps commute, which is to say that $A_i A_j = A_j A_i$ for every $i, j \in [n]$. In addition, we claim that with high probability, these maps are contractions (i.e., the associated matrices have operator norm at most 1). To see this, for each $i \in [n]$, let M_i be the random matrix given by $M_i = (T_{i,j,k})_{j,k=1}^n$. Observe that M_i is symmetric and its entries have mean zero and absolute value at most 1. By Lemma 4.5.3 and a union bound, we get that

$$\Pr\Big[\max_{i \in [n]} \|M_i\| > \tau\sqrt{n} \Big] \leq Cne^{-\kappa\tau n}. \tag{4.13}$$

for absolute constants κ, C and provided $\tau \geq C$.

Now, for every Euclidean unit vector $u \in \mathbb{R}^{2n+2}$, we have

$$\begin{aligned} \|A_i u\|_2^2 &= |u_0|^2 + \frac{1}{\tau^2 n} \sum_{k=1}^{n} \Big| \sum_{j=1}^{n} u_j T_{i,j,k} \Big|^2 + |u_{i+n}|^2 \\ &\leq |u_0|^2 + \frac{\|M_i\|^2}{\tau^2 n} \sum_{j=1}^{n} |u_j|^2 + |u_{i+n}|^2. \end{aligned} \tag{4.14}$$

It follows from Eq. (4.13) that $\max_i \|M_i\| \leq \tau\sqrt{n}$ with probability at least $1 - Cne^{-\kappa\tau n}$, which in turn implies that Eq. (4.14) is at most $\|u\|_2^2 \leq 1$ and therefore we have that all A_is have operator norm at most 1.

By Proposition 4.5.4,

$$\|p\|_{\mathrm{cb}} \geq \Big\| \sum_{i,j,k=1}^{n} T_{i,j,k} A_i A_j A_k \Big\|, \tag{4.15}$$

provided that the A_is are contractions.

By Eq. (4.12), and since $|T_{i,j,k}| \geq 1/6$ for every $i, j, k \in [n]$, the left hand side of Eq. (4.15) is at least $n^{5/2}/(36\tau)$, with probability at least $1 - Cne^{-\kappa\tau n}$. Letting τ be a sufficiently large constant then gives the result. □

As mentioned in the introduction, one can easily extend this result to the case of 4-linear forms. To demonstrate the failure of Theorem 4.1.1 for quartic polynomials, we embed our cubic polynomial into a quartic polynomial, which also gives a similar separation as in the cubic case.

4.5.5. Corollary. *There exists a bounded quartic form*

$$q(x_1, \ldots, x_n) = \sum_{\alpha \in \{0,1\}^n : |\alpha| = 4} d_\alpha x^\alpha, \tag{4.16}$$

and pairwise commuting contractions $A_1, \ldots, A_n \in L(\mathbb{R}^{2n+2})$ *such that*

$$\Big\| \sum_{i,j,k,\ell=1}^{n} (T_q)_{i,j,k,\ell} A_i A_j A_k A_\ell \Big\| \geq \kappa \sqrt{n}$$

where $\kappa \in (0, 1]$ *is some absolute constant.*

Proof. Let p be a bounded multi-linear cubic form such that $\|p\|_{\mathrm{cb}} \geq C\sqrt{n}$, the existence of which is guaranteed by Theorem 4.5.1. Let $T_p \in \mathbb{R}^{n \times n \times n}$ be the random symmetric 3-tensor associated to p. Consider the symmetric 4-tensor $S \in \mathbb{R}^{(n+1)\times(n+1)\times(n+1)\times(n+1)}$ defined by $S_{0,j,k,\ell} = T_{j,k,\ell}$, $S_{i,0,k,\ell} = T_{i,k,\ell}$, $S_{i,j,0,\ell} = T_{i,j,\ell}$, $S_{i,j,k,0} = T_{i,j,k}$ for every $i, j, k, \ell \in [n]$ and $S_{i,j,k,\ell} = 0$ otherwise. Since S is symmetric, there exists a unique multi-linear quartic form q associated to S. It follows easily that $\|q\| = 4\|p\|$. Moreover, by considering the contractions A_i used in the proof of Theorem 4.5.1 and defining $A_0 = \mathbf{1}_{n+2}$, it follows that $\|q\|_{\mathrm{cb}} \geq 4\|p\|_{\mathrm{cb}}$. The form $q/4$ is thus as desired. □

We claim that a form q as in Corollary 4.5.5 gives a counterexample to possible quartic extensions of Theorem 4.1.1. To see this, suppose there exists a two-query quantum algorithm $\mathcal{A}$ and a $C \in (0, \infty)$ (independent of n and $\mathcal{A}$) such that $\mathbb{E}[\mathcal{A}(x)] = Cq(x)$ for each $x \in \{-1, 1\}^n$. By Theorem 4.2.2, there exists a $(2n)$-variate quartic form h such that $h(x, 1^n) = Cq(x)$ for each $x \in \{-1, 1\}^n$ and $\|h\|_{\mathrm{cb}} \leq 1$. We now show that the degree-4 coefficients in $h(x, y)$ are completely determined by $q(x)$. Indeed, if we expand

$$h(x, y) = \sum_{\substack{\alpha,\beta \in \{0,1,2,3,4\}^n : \\ |\alpha|+|\beta|=4}} d'_{\alpha,\beta} x^\alpha y^\beta,$$

then, by the definition of q in Eq. (4.16), it follows that

$$\sum_{\substack{\alpha,\beta \in \{0,1,2,3,4\}^n : \\ |\alpha|+|\beta|=4}} d'_{\alpha,\beta} x^\alpha = h(x, 1^n) = Cq(x) = C \sum_{\alpha \in \{0,1\}^n : |\alpha| = 4} d_\alpha x^\alpha. \tag{4.17}$$

In particular, it follows from Eq. (4.17) that $d'_{\alpha,0^n} = Cd_\alpha$ for all $\alpha \in \{0, 1\}^n$ such that $|\alpha| = 4$.

In order to lower bound $\|h\|_{\mathrm{cb}}$, let $T_h \in \mathbb{R}^{(2n)\times(2n)\times(2n)\times(2n)}$ be the symmetric 4-tensor associated to h. By Proposition 4.5.4, we have

$$\|h\|_{\mathrm{cb}} \geq \Big\| \sum_{i,j,k,\ell=1}^{2n} (T_h)_{i,j,k,\ell} B_i B_j B_k B_\ell \Big\|,$$

for every set of pairwise commuting contractions $B_1, \dots, B_{2n}$. In particular, set $B_i = A_i$ as in Corollary 4.5.5 for $i \in [n]$ and let B_i be the all-zero matrix for $i \in \{n+1, \dots, 2n\}$. Since the A_is were pairwise commuting in Corollary 4.5.5 (which clearly commute with the all-zero matrix), the B_is are pairwise commuting. Finally, observe that for all $i, j, k, \ell \in [n]$, we have $(T_h)_{i,j,k,\ell} = d'_{\alpha,0^n}/(|\{i,j,k,\ell\}|!)$, which is equal to $Cd_\alpha/(|\{i,j,k,\ell\}|!)$ (by Eq. (4.17)). In particular, using Corollary 4.5.5, we have

$$\|h\|_{\mathrm{cb}} \geq \Big\| \sum_{i,j,k,\ell=1}^{2n} (T_h)_{i,j,k,\ell} B_i B_j B_k B_\ell \Big\| = C \Big\| \sum_{i,j,k,\ell=1}^{n} (T_q)_{i,j,k,\ell} A_i A_j A_k A_\ell \Big\| \geq C\kappa\sqrt{n}.$$

This implies that $1 \geq \|h\|_{\mathrm{cb}} = C\|q\|_{\mathrm{cb}} \geq C\kappa\sqrt{n}$, and so $C \leq 1/(\kappa\sqrt{n})$.

4.5.2 Upper bound on separation for cubic polynomials

In this section, we will prove that the separation we obtained in Theorem 4.5.1 is in fact optimal for degree-3 polynomials.

4.5.6. Theorem. *For every degree-3 polynomial $p : \mathbb{R}^n \to \mathbb{R}$, we have $\|p\|_{\mathrm{cb}} \leq \frac{63K_G}{2}\sqrt{n}\|p\|$, where K_G is Grothendieck's constant (see Theorem 4.5.8).*

Before we prove the theorem, we remark that a similar argument can also be used to show that for every quartic polynomial $p : \mathbb{R}^n \to \mathbb{R}$, we have $\|p\|_{\mathrm{cb}} \leq O(n\|p\|)$. We prove the degree-3 case for simplicity.

Our proof of the theorem uses the following polarization identity and the well-known Grothendieck theorem.

4.5.7. Lemma (Polarization identity [Tho14, Eq. 7]). *Let E, F be linear spaces and $T : E^n \to F$ be a symmetric multilinear map. Define p as*

$$p(x) = T(x, \dots, x) \quad \text{for all } x \in E.$$

Then, the polarization identity is given by:

$$T(x_1, \dots, x_n) = \frac{1}{n!} \sum_{k=1}^{n} (-1)^{n-k} \sum_{S \in \{0,1\}^n : |S| = k} p\Big(\sum_{i \in S} x_i \Big).$$

4.5.8. Theorem (Grothendieck). *There exists a universal constant $K_G \in (0, \infty)$ such that the following holds. For every positive integer n and matrix $A \in \mathbb{R}^{n \times n}$, we have*

$$\sup \Big\{ \sum_{i,j=1}^{n} A_{ij} \langle u_i, v_j \rangle : d \in \mathbb{N},\ \|u_i\|_2, \|v_j\|_2 \leq 1 \Big\} \leq K_G \, \|A\|_{\ell_\infty \to \ell_1}.$$

Elementary proofs of this theorem can be found for instance in [AN06]. The *Grothendieck constant* K_G is the smallest real number for which Theorem 4.5.8 holds true. The problem of determining its exact value, posed in [Gro53], remains open. The best lower and upper bounds $1.6769\cdots \leq K_G < 1.7822\cdots$ were proved by Davie and Reeds [Dav84, Ree91], and Braverman et al. [BMMN13], resp.

Proof of Theorem 4.5.6. The theorem follows immediately from the following lemma, which gives an upper bound on the completely bounded norm of trilinear forms in terms of their operator norm.

4.5.9. Lemma. *Let* $T = (T_{i,j,k})_{i,j,k=1}^{n}$ *be a sequence of* $\{-1,1\}$ *variables, then* $\|T\|_{\mathrm{cb}} \leq K_G\sqrt{n}\|T\|_{\ell_\infty,\ell_\infty,\ell_\infty}$.

Before we prove lemma 4.5.9, we conclude the proof of the theorem. Suppose T_p is the symmetric 3-linear form defined in Eq. (4.2). Observe that

$$\|p\|_{\mathrm{cb}} \leq \|T_p\|_{\mathrm{cb}}. \tag{4.18}$$

Indeed, since the definition of $\|p\|_{\mathrm{cb}}$ has an infimum over decompositions $T_p = \sum_\sigma T^\sigma \circ \sigma$, one possible decomposition is obtained by letting $T^\sigma = T_p/3!$ for all $\sigma \in S_3$. Note that this choice of T^σs is possible because T_p is symmetric. For this choice of T^σs, we get $\sum_{\sigma\in S_3}\|T^\sigma\|_{\mathrm{cb}} = \|T_p\|_{\mathrm{cb}}$, hence showing that $\|p\|_{\mathrm{cb}} \leq \|T_p\|_{\mathrm{cb}}$.

Using Lemma 4.5.9, we can now upper bound $\|T_p\|_{\mathrm{cb}} \leq K_G\sqrt{n}\|T_p\|_{\ell_\infty,\ell_\infty,\ell_\infty}$. Finally, in order to upper bound $\|T_p\|_{\ell_\infty,\ell_\infty,\ell_\infty}$, we use the polarization identity in Lemma 4.5.7 to first write

$$T_p(x_1,x_2,x_3) = \frac{1}{3!}\sum_{k=1}^{3}(-1)^{3-k}\sum_{\substack{S\in\{0,1\}^3:\\ |S|=k}} p\Big(\sum_{i\in S} x_i\Big).$$

Then, using the triangle inequality, it follows that

$$\begin{aligned}\|T_p\|_{\ell_\infty,\ell_\infty,\ell_\infty} &= \max_{x_1,x_2,x_3\in\{-1,1\}^n}|T_p(x_1,x_2,x_3)| \\ &\leq \frac{7}{6}\max_{x\in\{-3,\ldots,3\}^n}|p(x)| \\ &\leq \frac{7\cdot 27}{6}\max_{x\in\{-1,1\}^n}|p(x)| = \frac{63}{2}\|p\|,\end{aligned} \tag{4.19}$$

where we used that p is a degree-3 polynomial to conclude $\max_{x\in\{-\alpha,\ldots,\alpha\}^n}|p(x)| \leq \alpha^3\max_{x\in\{-1,1\}^n}|p(x)|$ for all $\alpha > 0$, in the second inequality.

Putting everything together, we get

$$\|p\|_{\mathrm{cb}} \overset{\text{Eq. (4.18)}}{\leq} \|T_p\|_{\mathrm{cb}} \overset{\text{Lemma 4.5.9}}{\leq} K_G\sqrt{n}\|T_p\|_{\ell_\infty,\ell_\infty,\ell_\infty} \overset{\text{Eq. (4.19)}}{\leq} \frac{63K_G}{2}\sqrt{n}\|p\|.$$

It remains to prove Lemma 4.5.9, which we do now.

Proof of Lemma 4.5.9. By definition of the completely bounded norm, our goal is to upper bound the following quantity

$$\|T\|_{\mathrm{cb}} = \Big\| \sum_{i,j,k=1}^{n} T_{i,j,k} A_i B_j C_k \Big\|,$$

where $\{A_i\}, \{B_j\}, \{C_k\}$ are sequences of matrices with operator norm at most 1. Suppose unit vectors u, v maximize the operator norm of $\sum_{i,j,k=1}^{n} T_{i,j,k} A_i B_j C_k$. Let $S_k = \sum_{ij} T_{i,j,k} B_j^* A_i^*$, $u_k = S_k u$ and $v_k = C_k v$, so that

$$\|T\|_{\mathrm{cb}} = \Big| \sum_{k=1}^{n} u_k^* v_k \Big|.$$

Let $x_1, \ldots, x_n$ be independent and identically distributed $\{-1, 1\}$-valued Bernoulli random variables. Then, we rewrite $\|T\|_{\mathrm{cb}}$ as follows,

$$\|T\|_{\mathrm{cb}} = \Big| \sum_{k=1}^{n} \langle u_k, v_k \rangle \Big| = \Big| \mathbb{E}_x \Big[\langle \sum_{k=1}^{n} u_k x_k, \sum_{\ell=1}^{n} v_\ell x_\ell \rangle \Big] \Big|, \tag{4.20}$$

where the second equality used $\mathbb{E}_x[x_i x_j] = \delta_{i,j}$ for every $i, j \in [n]$. Using the Cauchy-Schwarz inequality twice, we get

$$\begin{aligned} \Big| \mathbb{E}_x \Big[\langle \sum_{k=1}^{n} u_k x_k, \sum_{\ell=1}^{n} v_\ell x_\ell \rangle \Big] \Big| &\leq \mathbb{E}_x \Big[\| \sum_{k=1}^{n} u_k x_k \|_2 \| \sum_{\ell=1}^{n} v_\ell x_\ell \|_2 \Big] \\ &\leq \Big(\mathbb{E}_x \Big[\| \sum_{k=1}^{n} u_k x_k \|_2^2 \Big] \Big)^{1/2} \Big(\mathbb{E}_x \Big[\| \sum_{\ell=1}^{n} v_\ell x_\ell \|_2^2 \Big] \Big)^{1/2}. \end{aligned} \tag{4.21}$$

We now upper bound both terms in the final expression by

$$\underbrace{\Big(\mathbb{E}_x \Big[\| \sum_{k=1}^{n} u_k x_k \|_2^2 \Big] \Big)^{1/2}}_{\leq K_G \|T\|_{\ell_\infty, \ell_\infty, \ell_\infty}} \cdot \underbrace{\Big(\mathbb{E}_x \Big[\| \sum_{\ell=1}^{n} v_\ell x_\ell \|_2^2 \Big] \Big)^{1/2}}_{\leq \sqrt{n}} \leq K_G \sqrt{n} \|T\|_{\ell_\infty, \ell_\infty, \ell_\infty}. \tag{4.22}$$

For the first underbraced upper bound in Eq. (4.22), fix $x \in \{-1, 1\}^n$ and let $Q_{ij} = \sum_k T_{i,j,k} x_k$. Then,

$$\begin{aligned} \| \sum_{k=1}^{n} u_k x_k \|_2 &= \| \sum_{k=1}^{n} S_k x_k u \|_2 \\ &= \| \sum_{i,j,k=1}^{n} T_{i,j,k} x_k B_j^* A_i^* u \|_2 \\ &= \| \sum_{i,j=1}^{n} Q_{ij} B_j^* A_i^* u \|_2 \leq \| \sum_{i,j=1}^{n} Q_{ij} B_j^* A_i^* \|, \end{aligned} \tag{4.23}$$

where the last inequality is by definition of the operator norm and using that u is a unit vector. Suppose w, z maximize the final norm expression, then

$$\|\sum_{i,j=1}^{n} Q_{ij} B_j^* A_i^*\| = |\sum_{i,j=1}^{n} Q_{ij} \langle B_j w, A_i^* z\rangle| \leq K_G \max_{y,z\in\{-1,1\}^n} \Big|\sum_{i,j=1}^{n} Q_{ij} y_i z_j\Big|, \quad (4.24)$$

where the last inequality used Grothendieck's Theorem 4.5.8. Putting together Eq. (4.23), (4.24) into Eq. (4.21), it now follows that

$$\begin{aligned}\Big(\mathbb{E}_x\Big[\|\sum_{k=1}^{n} u_k x_k\|_2^2\Big]\Big)^{1/2} &\leq \Big(\max_x \Big[\|\sum_{k=1}^{n} u_k x_k\|_2^2\Big]\Big)^{1/2} \\ &= \max_{x\in\{-1,1\}^n} \|\sum_{k=1}^{n} u_k x_k\|_2 \\ &\overset{\text{Eq. (4.24,4.25)}}{\leq} K_G \max_{x,y,z\in\{-1,1\}^n} \sum_{i,j=1}^{n} Q_{ij} y_i z_j = K_G \|T\|_{\ell_\infty,\ell_\infty,\ell_\infty}.\end{aligned} \quad (4.25)$$

For the second underbraced upper bound in Eq. (4.22), observe that

$$\begin{aligned}\mathbb{E}_x\Big[\|\sum_{\ell=1}^{n} v_\ell x_\ell\|_2^2\Big] = \mathbb{E}_x\Big[\sum_{\ell,\ell'=1}^{n} v_\ell x_\ell v_{\ell'} x_{\ell'}\Big] &= \sum_{\ell=1}^{n} \|v_\ell\|_2^2 \\ &= \sum_{\ell=1}^{n} \|C_\ell v\|_2^2 \leq \sum_{\ell=1}^{n} \|C_\ell\|^2 \leq n,\end{aligned} \quad (4.26)$$

where we used $\mathbb{E}_x[x_i x_j] = \delta_{i,j}$ in the first equality, the definition of $v_\ell = C_\ell v$ in the second equality and in the last inequality used that the C_ℓs have operator norm at most 1. Putting together Eq. (4.20), Eq. (4.21) and Eq. (4.22), we have

$$\|T\|_{\text{cb}} \leq K_G \sqrt{n} \|T\|_{\ell_\infty,\ell_\infty,\ell_\infty},$$

concluding the proof of the lemma. □

□

4.6 Short proof of Theorem 4.1.1

In this section, we give a short proof of Theorem 4.1.1, restated below for convenience.

4.6.1. Theorem (Aaronson et al.). *There exists an absolute constant $C \in (0, 1]$ such that the following holds. For every bounded quadratic polynomial p, there exists a one-query quantum algorithm that, on input $x \in \{-1, 1\}^n$, returns a random sign with expectation $Cp(x)$.*

We begin by giving a brief sketch of the original proof.

Proof sketch of Theorem 4.1.1. The first step is to show that without loss of generality, we may assume that the polynomial p is a quadratic form. This is the content of the decoupling argument mentioned in the introduction, proved for polynomials of arbitrary degree in [AAI+16], but stated here only for the quadratic case.

4.6.2. Lemma. *There exists an absolute constant $C \in (0,1]$ such that the following holds. For every bounded quadratic polynomial p, there exists a matrix $A \in \mathbb{R}^{(n+1)\times(n+1)}$ with $\|A\|_{\ell_\infty \to \ell_1} \leq 1$, such that the quadratic form $q(y) = y^\mathsf{T} A y$ satisfies $q((x,1)) = Cp(x)$ for all $x \in \{-1,1\}^n$.*

To prove the theorem, we may thus restrict to a quadratic form $p(x) = y^\mathsf{T} A y$ given by some matrix $A \in \mathbb{R}^{(n+1)\times(n+1)}$ such that $\|A\|_{\ell_\infty \to \ell_1} \leq 1$. The next step is to massage the matrix A into a unitary matrix (that can be applied by a quantum algorithm). To obtain this unitary, the authors use an argument based on two versions of Grothendieck's inequality and a technique known as *variable splitting*, developed in earlier work of Aaronson and Ambainis [AA15]. The first version of Grothendieck's inequality is the one most commonly used in applications [Gro53] and stated in Theorem 4.5.8.

The second version of Grothendieck's inequality is as follows.

4.6.3. Theorem (Grothendieck). *For every positive integer n and matrix $A \in \mathbb{R}^{(n+1)\times(n+1)}$, there exist $u, v \in (0,1]^{(n+1)}$ such that $\|u\|_2 = \|v\|_2 = 1$ and such that the matrix*

$$B = \frac{1}{K_G} \operatorname{Diag}(u)^{-1} A \operatorname{Diag}(v)^{-1} \tag{4.27}$$

satisfies $\|B\| \leq \|A\|_{\ell_\infty \to \ell_1}$, where $\operatorname{Diag}(w)$ denotes the square diagonal matrix whose diagonal is w.

4.6.1 Our contribution.

The first (standard) version of Grothendieck's inequality (Theorem 4.5.8) easily implies that every matrix A such that $\|A\|_{\ell_\infty \to \ell_1} \leq 1$ has completely bounded norm at most K_G. Combing this fact with our Theorem 4.2.2 and Lemma 4.6.2, one quickly retrieves Theorem 4.1.1. However, Theorem 4.2.2 is based on the rather deep Theorem 4.4.1. We observe that Theorem 4.1.1 also follows readily from the much simpler Theorem 4.6.3 alone (proved below for completeness), after one assumes that q is a quadratic form as above.

Indeed, Theorem 4.6.3 gives unit vectors u, v such that the matrix B as in Eq. (4.27) has (operator) norm at most 1. Unitary matrices have norm exactly 1 and of course represent the type of operation a quantum algorithm can implement. Moreover, since u, v are unit vectors, they represent $(\log(n+1))$-qubit quantum states. Using the fact that for $w, z \in \mathbb{R}^{(n+1)}$, we have $\operatorname{Diag}(w)z = \operatorname{Diag}(z)w$, we

get the following *factorization* formula (not unlike the one of Corollary (4.4.2), which is of course no coincidence):

$$\frac{y^{\mathsf{T}}Ay}{K_G} = y^{\mathsf{T}}\operatorname{Diag}(u)B\operatorname{Diag}(v)y = u^T\operatorname{Diag}(y)B\operatorname{Diag}(y)v. \tag{4.28}$$

If we assume for the moment that the matrix B actually is unitary, then the right-hand side of Eq. (4.28) suggests the simple one-query quantum algorithm described in Figure 4.2.

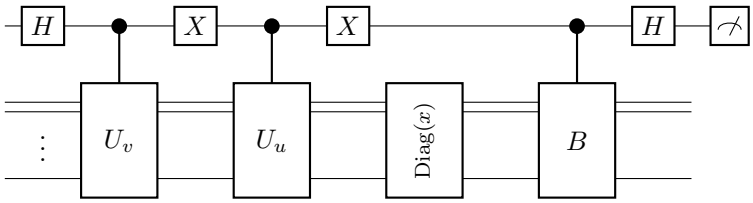


Figure 4.2: Let U_u, U_v be unitaries that have u, v as their first rows, respectively and for $x \in \{-1,1\}^n$ and $y = (x,1)$, let $\operatorname{Diag}(y)$ be the query operator. The algorithm initializes a $(1+\log(n+1))$-qubit register in the all-zero state, transforms this state into the superposition $\frac{1}{\sqrt{2}}(|0\rangle \otimes u + |1\rangle \otimes v)$, makes a query via the unitary $\operatorname{Diag}(y)$ applied to the $(\log(n+1))$-qubit register, applies a controlled-B, and finishes by measuring the first qubit in the Hadamard basis.

$$\frac{1}{2} + \frac{1}{2}\big\langle \operatorname{Diag}(u)y, B\operatorname{Diag}(v)y\big\rangle = \frac{1}{2} + \frac{y^{\mathsf{T}}Ay}{2K_G},$$

Now, it is clear that the the expected value of the measurement result is precisely $q((x,1))/K_G$, giving Theorem 4.1.1 with $C = 1/K_G$. In case B is not unitary, one can use the same argument as in the final step of the proof of Theorem 4.2.2.

4.6.2 Factorization version of Grothendieck's inequality

For completeness and because of its relevance to Theorem 4.1.1, we here give a proof of Theorem 4.6.3. The proof relies on the standard version of Grothendieck's inequality (Theorem 4.5.8). In addition, the proof makes use of the following version of the Hahn–Banach theorem [Rud91, Theorem 3.4].

4.6.4. Theorem (Hahn–Banach separation theorem). *Let $C, D \subseteq \mathbb{R}^n$ be convex sets and let C be algebraically open. Then the following are equivalent:*

- *The sets C and D are disjoint.*

- *There exists a vector $\lambda \in \mathbb{R}^n$ and a constant $\alpha \in \mathbb{R}$ such that $\langle \lambda, d\rangle > \alpha$ for every $d \in D$ and $\langle \lambda, c\rangle \le \alpha$ for every $c \in C$.*

Morever, if C and D are convex cones,[6] *we may take $\alpha = 0$.*

Proof of Theorem 4.6.3. Let $M = A/(K_G \|A\|_{\ell_\infty \to \ell_1})$. By Theorem 4.5.8 (the standard Grothendieck inequality), we have that

$$\sum_{i,j=1}^{n} M_{ij}\langle x_i, y_j\rangle \le 1$$

for all vectors x_i, y_j with Euclidean norm at most 1. Then, for arbitrary vectors x_i, y_j, we have

$$\sum_{i,j=1}^{n} M_{ij}\langle x_i, y_j\rangle \le \max_{i,j\in[n]} \|x_i\|_2 \|y_j\|_2 \le \frac{1}{2} \max_{i,j\in[n]} (\|x_i\|_2^2 + \|y_j\|_2^2), \tag{4.29}$$

where the second inequality is by AM-GM inequality. Define the set $K \subseteq \mathbb{R}^{n\times n}$ by

$$K = \left\{ \left(\|x_i\|_2^2 + \|y_j\|_2^2 - 2\sum_{k,\ell=1}^{n} M_{k\ell}\langle x_k, y_\ell\rangle \right)_{i,j=1}^{n} : d \in \mathbb{N},\ \ x_i, y_j \in \mathbb{R}^d \right\}.$$

We claim that K is a convex cone. Observe that for every $t > 0$ and matrix $Q \in K$ specified by the set of vectors $\{x_i\}, \{y_j\}$, the vectors $x_i' = \sqrt{t}x_i$ and $y_j' = \sqrt{t}y_j$ similarly define tQ, and so K is a cone. We now show K is a convex set. Let $Q, Q' \in K$ be specified by x_i, y_j and x_i', y_j' respectively. Then, for every $\lambda \in [0,1]$, the convex combination $\lambda Q + (1-\lambda)Q'$ also belongs to K, as it can be specified by the vectors $(\sqrt{\lambda}x_i, \sqrt{1-\lambda}x_i'), (\sqrt{\lambda}y_j, \sqrt{1-\lambda}y_j')$.

Additionally, it follows from Eq. (4.29) that K is disjoint from the open convex cone $\mathbb{R}^{n\times n}_{<0}$ of matrices with strictly negative entries. By Theorem 4.6.4 (the Hahn–Banach separation theorem), we conclude that there exists a nonzero matrix $L \in \mathbb{R}^{n\times n}$ such that $\langle L, Q\rangle > 0$ for every $Q \in K$ and $\langle L, N\rangle \le 0$ for every $N \in \mathbb{R}^{n\times n}_{<0}$. In particular, the second inequality implies that $L_{ij} > 0$ for every $i, j \in [n]$. Let $P = L/\sum_{ij} L_{ij}$, so that $\{P_{ij}\}_{i,j=1}^n$ defines a probability distribution over $[n]^2$. Then, for every $Q \in K$,

$$\begin{aligned} 0 &\le \langle P, Q\rangle \\ &= \sum_{i,j=1}^{n} P_{ij}(\|x_i\|_2^2 + \|y_j\|_2^2) - 2\sum_{k,\ell=1}^{n} M_{k\ell}\langle x_k, y_\ell\rangle \\ &= \sum_{i=1}^{n} \sigma_i \|x_i\|_2^2 + \sum_{j=1}^{n} \mu_j \|y_j\|_2^2 - 2\sum_{k,\ell=1}^{n} M_{k\ell}\langle x_k, y_\ell\rangle, \end{aligned}$$

[6] A convex cone $\mathcal{K}$ is a set that satisfies: (i) for every $x \in \mathcal{K}$ and $\lambda > 0$, we have $\lambda x \in \mathcal{K}$ and (ii) for every $x, y \in \mathcal{K}$, we have $x + y \in \mathcal{K}$.

where $\sigma_i = P_{i1} + \cdots + P_{in}$ and $\mu_j = P_{1j} + \cdots + P_{nj}$. Observe that σ_i, μ_j are strictly positive because $P_{ij} > 0$. Rearranging the inequality above and using bi-linearity, it follows that for every $\lambda > 0$, we have

$$\begin{aligned} 2\sum_{k,\ell=1}^{n} M_{k\ell}\langle x_k, y_\ell\rangle &= 2\sum_{k,\ell=1}^{n} M_{k\ell}\langle \lambda x_k, \lambda^{-1} y_\ell\rangle \\ &\leq \lambda^2 \sum_{i=1}^{n} \sigma_i \|x_i\|_2^2 + \lambda^{-2}\sum_{j=1}^{n} \mu_j \|y_j\|_2^2. \end{aligned} \tag{4.30}$$

Setting

$$\lambda = \left(\frac{\sum_{j=1}^{n} \mu_j \|y_j\|_2^2}{\sum_{i=1}^{n} \sigma_i \|x_i\|_2^2}\right)^{1/4}$$

in Eq. (4.30), we find that

$$2\sum_{k,\ell=1}^{n} M_{k\ell}\langle x_k, y_\ell\rangle \leq \Big(\sum_{i=1}^{n} \sigma_i \|x_i\|_2^2\Big)^{1/2} \Big(\sum_{j=1}^{n} \mu_j \|y_j\|_2^2\Big)^{1/2}.$$

In particular, for the case where $x_k, y_\ell \in \mathbb{R}$, i.e., the scalar case, we have

$$x^{\mathsf{T}} M y \leq \|\operatorname{diag}(\sigma)^{1/2} x\|_2 \|\operatorname{diag}(\mu)^{1/2} y\|_2.$$

This implies

$$x^{\mathsf{T}} \Big(\operatorname{Diag}(\sigma)^{-1/2} M \operatorname{Diag}(\mu)^{-1/2}\Big) y \leq \|x\|_2 \cdot \|y\|_2,$$

which in particular implies that $\|\operatorname{Diag}(\sigma)^{-1/2} M \operatorname{Diag}(\mu)^{-1/2}\| \leq 1$. Using the definition of $M = A/(K_G \|A\|_{\ell_\infty \to \ell_1})$, we have

$$\|\operatorname{Diag}(\sigma)^{-1/2} A \operatorname{Diag}(\mu)^{-1/2}\| \leq K_G \|A\|_{\ell_\infty \to \ell_1}.$$

The theorem follows by letting $u_i = \sqrt{\sigma_i}$, $v_i = \sqrt{\mu_i}$ for every $i \in [n]$. □

4.7 Conclusion and future work

In this chapter, we refined the polynomial method by defining a new degree measure, called the completely bounded approximate degree of a Boolean function f, that equals the quantum query complexity of f. Thereby, our characterization in Corollary 4.2.4 allows us to give upper bounds on quantum query complexity in terms of a degree measure. Prior to this work, the quantum adversary method was used as a generic technique to construct quantum algorithms [Rei09, LMR+11, Bel12, Kim13, LL16]. However, showing bounds on the

generalized adversary method seems hard in general. In contrast, our characterization bypasses the need of the adversary method and gives a potentially new perspective on constructing quantum algorithms. As a first step, could we recover results in quantum query complexity such as: Grover's search algorithm [Gro96], Ambainis's algorithm for element distinctness [Amb07], or the algorithm for NAND tree evaluation [ACR+10]. Using the adversary method, Reichardt [Rei11] showed that quantum query complexity composes i.e., $Q_\varepsilon(f \circ g) = \Theta(Q_\varepsilon(f)Q_\varepsilon(g))$,[7] Using our characterization this also implies that $\text{cb-deg}_\varepsilon(f \circ g) = \Theta(\text{cb-deg}_\varepsilon(f)\text{cb-deg}_\varepsilon(g))$, which was unknown in operator space theory, as far as we know. An interesting question is, can we show $\text{cb-deg}_\varepsilon(f \circ g) = \Theta(\text{cb-deg}_\varepsilon(f)\text{cb-deg}_\varepsilon(g))$ *without* using Reichardt's result? This will reprove the already-known and important result that quantum query complexity composes! We remark that if $\text{cb-deg}_\varepsilon(f)$ is replaced with $\deg_\varepsilon(f)$, then by a result of Sherstov [She13], we know that $\deg_\varepsilon(f \circ g) \leq O(\deg_\varepsilon(f)\deg_\varepsilon(g))$. It is an open question if $\deg_\varepsilon(f \circ g) \geq \Omega(\deg_\varepsilon(f)\deg_\varepsilon(g))$ for all total Boolean functions f, g.

[7] Here $f \circ g : \{-1,1\}^{n^2} \to \{-1,1\}$ is defined as $(f \circ g)(x^1, \ldots, x^n) = f(g(x^1), \ldots, g(x^n))$ for every $x^i \in \{-1,1\}^n$.

Chapter 5

Quantum gradient-based optimization

This chapter is based on the paper "Optimizing quantum optimization algorithms via faster quantum gradient computation", by A. Gilyén, S. Arunachalam and N. Wiebe [GAW17].

Abstract. Optimization is a fundamentally important task that touches on virtually every area of science. Quantum algorithms are known to provide substantial improvements for several related problems [Gro96, Jor05, HHL09, CKS15, BS17, AGGW17]. However, applying non-Grover techniques to real-word optimization problems has proven challenging, because generic problems usually fail to satisfy the delicate requirements of these advanced quantum techniques.

In this chapter, we look at continuous-variable optimization problems and in particular, we provide a quantum speed-up for gradient-based optimization methods. We develop a quantum algorithm for computing the gradient of a multivariate function that improves upon Jordan's quantum algorithm [Jor05]. Our quantum algorithm is quadratically better than classical gradient computation algorithms in terms of query and time complexity (under reasonable continuity assumptions). Furthermore, we use our improved gradient computation algorithm to improve the complexity of most gradient-based optimization algorithms. Since gradient-based optimization is ubiquitous in classical machine learning, our quantum improvement improves upon almost all gradient-based machine learning algorithms. Finally we briefly mention some results in [GAW17] whose details are omitted in this chapter.

Contents

5.1 Introduction

The last two decades have seen many quantum algorithms for computational problems in number theory [Sho97], search problems [Gro96], formula evaluation [ACR+10], Hamiltonian simulation [BCK15], solving linear systems [HHL09] and machine learning tasks [WKS15, WKS16b].[1] However, less attention has been devoted to developing quantum algorithms for discrete and continuous optimization problems which are possibly intractable by classical computers. Naïvely, since Grover's quantum algorithm [Gro96] quadratically improves upon the classical algorithm for searching in a database, we can simply use it to speed up all discrete optimization algorithms which involve searching for a solution among a set of unstructured possible solutions. However, in real-world applications, many problems have *continuous* parameters, where an alternative quantum optimization approach might fit the problem better.

A handful of quantum algorithms for specific continuous-variable optimization problems were developed for quantum adiabatic optimization [FGGS00], quantum annealing [KN98], Monte Carlo methods [Mon15], derivative-free optimization [Aru14], least squares fitting [WBL12], optimization algorithms for satisfiability and travelling salesman problem [HP00, Aru14] and quantum approximate optimization [FGG14]. Also, very recently, there has been work on quantum algorithms for solving linear and semi-definite programs [BS17, AGGW17, BKL+17].

[1] See the "Quantum Algorithm Zoo": http://math.nist.gov/quantum/zoo/ for a comprehensive list of quantum algorithms for computational problems.

In this chapter, we consider *gradient-based optimization*, which is a well-known technique to handle continuous-variable optimization problems. In particular we focus on the *gradient-descent algorithm*, which is a first-order optimization algorithm[2] used to find the minimum of a multivariate function $f : \mathbb{R}^d \to \mathbb{R}$. This optimization technique is used often in machine learning applications such as neural networks, support vectors machines and regression. In this direction, Jordan [Jor05] constructed a quantum algorithm that computes the gradient of f using a single query, however he assumes an unusually strong access model, which seems unrealistic in practice. In [GAW17], we considered a more realistic access model and developed an improved quantum algorithm for gradient computation, which has query and gate complexity $\widetilde{O}(\sqrt{d})$ (under reasonable continuity assumptions) for most functions. We remark here that our $d \to \sqrt{d}$ speed-up doesn't come by simply applying Grover's algorithm, instead our speed-up comes from the quantum Fourier transform. Using this gradient-calculation algorithm as a subroutine, we improved the complexity of most classical gradient-descent algorithms.

Organization. In Section 5.2, we begin by describing the classical gradient-based optimization algorithm and describe our quantum improvements to this classical algorithm. In Section 5.3, we formally analyze Jordan's algorithm and its complexity and then present our quantum improvements to his algorithm. In Section 5.4, we describe a few other results present in [GAW17] which are not present in this chapter. We conclude in Section 5.5 with some directions for future research.

5.2 Gradient-based optimization

5.2.1 Classical gradient-based optimization algorithms

In this section, we give a brief description of a classical gradient-based algorithm for optimization. Consider the multivariate function $p : \mathbb{R}^d \to \mathbb{R}$ and assume for simplicity that p is bounded by some absolute constant (i.e., there exists a universal constant $C > 0$ such that $|p(\boldsymbol{x})| \leq C$ for all $\boldsymbol{x}$) and differentiable everywhere. The optimization problem we are interested in is, given $p : \mathbb{R}^d \to \mathbb{R}$, compute

$$\mathrm{OPT} = \min\{p(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^d\}. \tag{5.1}$$

First-order iterative methods are optimization techniques that compute an approximation to OPT, simply using the gradient of p,

$$\nabla p = \left(\frac{\partial p}{\partial x_1}, \frac{\partial p}{\partial x_2}, \ldots, \frac{\partial p}{\partial x_d}\right) \tag{5.2}$$

[2]Gradient-descent algorithm is referred to as a *first-order* method because the algorithm uses *only* the first derivative (i.e., gradient) of the objective function that we are trying to optimize.

It is a well-known fact in calculus that p decreases the *fastest* in the direction of $-(\nabla p(\boldsymbol{x}))$. This simple observation is the basis of gradient-descent optimization algorithms.

Now we describe a simple heuristic gradient-descent algorithm for approximating (5.1): pick a random point $\boldsymbol{x}^{(0)} \in \mathbb{R}^d$, compute $\nabla p(\boldsymbol{x}^{(0)})$, and take a δ_0-step in the direction of $-\nabla p(\boldsymbol{x}^{(0)})$ leading to $\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} - \delta_0 \nabla p(\boldsymbol{x}^{(0)})$ (for some step size $\delta_0 > 0$). Repeat this gradient update for T steps (possibly using different step sizes δ in each step), obtaining $\boldsymbol{x}^{(T)}$ which has hopefully approached some local minimum of (5.1). Finally repeat the whole procedure for N different starting points $\{\boldsymbol{x}_1^{(0)}, \ldots, \boldsymbol{x}_N^{(0)}\}$ and take the minimum of $\{p(\boldsymbol{x}_1^{(T)}), \ldots, p(\boldsymbol{x}_N^{(T)})\}$ after T gradient steps, hoping to have a good approximation of OPT. Suppose p is a convex function, then all local minima are also global minima, so this algorithm will converge to the global minimum. However, if p is not convex, there is a whole subject related to the topic of convergence analysis of gradient-based method, which we do not discuss here. Given the generality of the optimization problem (5.1) and the simplicity of this heuristic algorithm, gradient-based techniques are widely used in mathematics, physics and engineering. In practice, especially for well-behaved functions p (such as convex functions), gradient-based algorithms are known to converge very quickly to a global optimum and are often used, e.g., in state-of-the-art algorithms for deep learning [Rud16], which has been one of the recent highlights in classical machine learning.

5.2.2 Complexity measure and quantum sampling

In this chapter we consider if one can quantumly improve the classical gradient-descent algorithm described in the previous section. Clearly a key component of the gradient-descent algorithm is the computation of the gradient of p. In this chapter, we will be interested in quantumly improving the complexity of gradient computation. In this direction, Jordan [Jor05] described a quantum algorithm that computes the gradient of p using a *single* quantum query. However, his algorithm assumes that p is close to being linear and he also assumes an unusually strong oracle access to p. Roughly speaking, Jordan assumes access to a binary oracle that on input $\boldsymbol{x}$, outputs $p(\boldsymbol{x})$ with a good accuracy (we discuss this in detail later).

The starting point of our work was the simple realization that it is uncommon to assume access to the *binary oracle* in the applications of the gradient-descent algorithm for optimization problems. We observed that most classical optimization procedures evaluate p via sampling, i.e., these procedures output a bit which is '1' with probability $p(\boldsymbol{x})$. Using this sampling model, we first analyze the classical complexity of the gradient-descent algorithm described earlier.

It is not hard to see that using empirical estimation it suffices to use $O(1/\varepsilon^2)$ samples in order to evaluate $p(\boldsymbol{x})$ with additive error $\Theta(\varepsilon)$. Provided that p is

smooth[3] we can compute an ε-approximation of $\nabla_i p(\boldsymbol{x}) = \frac{\partial p}{\partial x_i}$ by performing $\widetilde{O}(1)$ such function evaluations, using standard classical techniques. Hence, we can compute an ε-approximation of the gradient $\nabla p(\boldsymbol{x})$ with $\widetilde{O}(d)$ function evaluations of precision $\Theta(\varepsilon)$. The simple gradient-descent algorithm described in the previous section uses TN gradient computations, therefore the overall algorithm can be executed using $\widetilde{O}(TNd/\varepsilon^2)$ samples.

We then observed that quantum optimization procedures translate the objective function (i.e., p) to the probability of some measurement outcome. To reflect this fact, we use an oracular model to represent our objective function that is much weaker (but more realistic) than the oracle model considered by Jordan [Jor05]. Here we work with a coherent version of the classical random sampling procedure, i.e., we assume that the function is given by a *probability oracle*:

$$U_p : |\boldsymbol{x}\rangle|0\rangle \mapsto \sqrt{p(\boldsymbol{x})}|\boldsymbol{x}\rangle|1\rangle + \sqrt{1-p(\boldsymbol{x})}|\boldsymbol{x}\rangle|0\rangle \quad \text{for every } \boldsymbol{x}, \tag{5.3}$$

where the continuous input variable $\boldsymbol{x}$ is represented as a finite-precision binary encoding of $\boldsymbol{x}$. In this chapter, we first address the question:

> How many queries to U_p suffice to compute the *gradient* of p?

5.2.3 Prior work on quantum gradient methods

Our gradient computation algorithm is based on Jordan's [Jor05] quantum algorithm, which provides an "exponential" quantum speed-up for gradient computation in a black-box model. However, as mentioned earlier, Jordan's algorithm assumes an unusually strong oracle access model. Bulger [Bul05a] later showed how to combine Jordan's algorithm with quantum minimum finding [DH96] to improve gradient-descent methods.

Recently, Rebentrost et al. [RSPL16] and Kerenidis and Prakash [KP17a] considered a very different approach, where they represent vectors as quantum states, which can lead to exponential improvements in terms of the dimension for *specific* gradient-based algorithms. Rebentrost et al. [RSPL16] obtained speed-ups for first and second-order iterative methods (i.e., gradient-descent and Newton's method) for polynomial optimization problems. The runtime of their quantum algorithm achieves poly-logarithmic dependence on the dimension d but scales exponentially with the number of gradient steps T. Kerenidis and Prakash [KP17a] described a gradient-descent algorithm for the special case of quadratic optimization problems. The algorithm's runtime scales linearly with the number of steps T and in some cases can achieve poly-logarithmic dependence on the dimension d as it essentially implements a version of the HHL algorithm [HHL09] for solving

[3] Throughout the chapter, when we say that a function is *smooth* we mean that it is analytic and has bounded partial derivatives. We formally define smooth in the preliminaries in Section 5.3.1.

linear systems. However, their appealing runtime bound requires a very strong access model for the underlying data. We remark that, in contrast to these earlier papers on gradient-based optimization, our algorithm outputs a classical description of the gradient.

5.2.4 Quantum speed-ups for the simple gradient-descent algorithm

Improved gradient calculation algorithm. Jordan's algorithm for gradient computation [Jor05] uses a *fairly strong* input model, it assumes that $p : \mathbb{R}^d \to \mathbb{R}$ is given by an η-accurate binary oracle, which on input $\boldsymbol{x}$, outputs $p(\boldsymbol{x})$ in binary with additive error η.[4] Using this oracle, Jordan showed how to compute an ε-coordinate-wise approximation of ∇p using a *single* evaluation of the binary oracle. The algorithm prepares a uniform superposition of evaluation points over a finite region, then approximately implements the $S = O(\sqrt{d}/\varepsilon^2)$-th power of a phase oracle

$$\mathrm{O}_p^S : |\boldsymbol{x}\rangle \to e^{iSp(\boldsymbol{x})}|\boldsymbol{x}\rangle,$$

using a *single* $\eta = \Theta(\varepsilon^2/\sqrt{d})$-accurate evaluation of p, and then applies an inverse Fourier transformation to obtain an approximation of the gradient. Although this algorithm only uses a single query, the required precision of the function evaluation can be prohibitive. In particular, if were to use quantum amplitude estimation [BHMT02] to implement the binary oracle, then the algorithm would make $\widetilde{O}(\sqrt{d}/\varepsilon^2)$ probability oracle queries to evaluate the function with accuracy $\eta = \Theta(\varepsilon^2/\sqrt{d})$. In contrast, our new quantum algorithm requires only $\widetilde{O}(\sqrt{d}/\varepsilon)$ queries to a probability oracle. The precise statement can be found in Theorem 5.3.4, and below we give an informal statement.

5.2.1. Theorem (Informal). *There is a quantum algorithm, that given probability oracle U_p (in Eq. (5.3)) access to an analytic function $p : \mathbb{R}^d \to [0, 1]$ having bounded partial derivatives at $\mathbf{0}$, computes an approximate gradient $\boldsymbol{g} \in \mathbb{R}^d$ such that $\|\boldsymbol{g} - \nabla p(\mathbf{0})\|_\infty \leq \varepsilon$ with high probability, using $\widetilde{O}(\sqrt{d}/\varepsilon)$ queries and elementary gates. We get similar complexity bounds if we are given phase oracle access to the function.*

Proof sketch. The main new ingredient of our algorithm is the use of higher-degree central-difference formulas, a technique borrowed from calculus. We use the fact that for a one-dimensional analytic function $h : \mathbb{R} \to \mathbb{R}$ having bounded derivatives at 0, there exists a $\log(1/\varepsilon)$-degree central-difference formula to compute an $\varepsilon \cdot \log(1/\varepsilon)$-approximation of $h'(0)$ using ε-accurate evaluations of h at

[4] This input model captures functions that are evaluated numerically using, say, an arithmetic circuit. Typically, the number of one- and two-qubit gates needed to evaluate such functions up to n digits precision is polynomial in n and d.

$\log(1/\varepsilon)$ different points around 0. We apply this result to one-dimensional slices of the d-dimensional function $p : \mathbb{R}^d \to \mathbb{R}$. The main technical challenge in our proof is to show that if p is smooth, then for most such one-dimensional slices, the k-th order directional derivatives increase by at most an $O((\sqrt{d})^k)$-factor compared to the partial derivatives of p. As we show this implies that it is enough to evaluate the function p with $\widetilde{O}(\varepsilon/\sqrt{d})$-precision in order to compute the gradient. After the function evaluations, our algorithm ends by applying a d-dimensional quantum Fourier transform providing a classical description of an approximate gradient, similarly to Jordan's algorithm. □

Improving gradient-based algorithm. Using our quantum gradient calculation algorithm, we briefly describe how to improve the simple gradient-descent algorithm described in Section 5.2.1. As discussed in Section 5.2.2, we assume that we have access to a probability oracle (5.3) and for simplicity, let us assume that the objective function p is smooth. First, we improve the complexity of ε-accurate function evaluations to $O(1/\varepsilon)$ using amplitude estimation [BHMT02]. Then, similar to [Bul05a, LPL14], we improve the parallel search for finding a global minimum using the quantum minimum-finding algorithm [DH96, AGGW17]. Additionally, using our quantum algorithm for gradient computation, we get a quadratic improvement in terms of the dimension d. In particular, this shows that we can speed up the gradient-based optimization algorithm quadratically in almost all parameters, except the number of iterations T. The results are summarized below in Table 5.1:

Method:	Classical algorithm	+Amplitude estimation	+Grover search	+**This chapter**
Complexity:	$\widetilde{O}(TNd/\varepsilon^2)$	$\widetilde{O}(TNd/\varepsilon)$	$\widetilde{O}(T\sqrt{N}d/\varepsilon)$	$\widetilde{O}(T\sqrt{Nd}/\varepsilon)$

Table 5.1: Quantum speed-ups for a simple gradient-descent algorithms

Remark about T. Since gradient-descent is ubiquitous in optimization, it has been optimized extensively in the *classical* literature, yielding significant reductions in the number of steps T, see for example accelerated gradient methods [Nes83, BT09, JKK+17]. We think it should be possible to combine some of these classical results with our quantum speed-up, because our algorithm outputs a classical description of the gradient, unlike other recent developments on quantum gradient-descent methods [RSPL16, KP17a]. However, there could be some difficulty in applying classical acceleration techniques, because they often require unbiased samples of the approximate gradient, which might be difficult to achieve using quantum sampling.

5.3 Quantum gradient calculation algorithm

5.3.1 Preliminaries

We quickly recap some notation that we need for this section and give a quick overview about the oracles that we encounter often in this section.

Notation. We use bold letters for vectors $\boldsymbol{x} \in \mathbb{R}^d$. For a set of vectors $S \subseteq \mathbb{R}^d$, we let $\boldsymbol{y} + rS = \{\boldsymbol{y} + r\boldsymbol{v} : \boldsymbol{v} \in S\}$. For $\boldsymbol{x} \in \mathbb{R}^d$, let $\|\boldsymbol{x}\|_\infty = \max_{i\in[d]} |x_i|$ and $\|\boldsymbol{x}\| = (\sum_{i=1}^d x_i^2)^{1/2}$. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, let $\boldsymbol{x} \cdot \boldsymbol{y} = \sum_{i=1}^d x_i y_i$. For $M \in \mathbb{R}^{d\times d}$ we use $\|M\|$ for the operator norm of M.

We will need the following definitions in higher-order calculus.

5.3.1. Definition (Index-sequences). For $k > 0$, we refer to $\alpha \in [d]^k$ as a d-dimensional length-k index-sequence. For a vector $\boldsymbol{r} \in \mathbb{R}^d$ we define $\boldsymbol{r}^\alpha := \prod_{j\in[k]} r_{\alpha_j}$. Also, for a k-times differentiable function, we define

$$\partial_\alpha f := \partial_{\alpha_1}\partial_{\alpha_2}\cdots\partial_{\alpha_k} f.$$

5.3.2. Definition (Analytic function). We say that the function $f : \mathbb{R}^d \to \mathbb{R}$ is analytic if f can be written as

$$f(\boldsymbol{x}) = \sum_{k=0}^{\infty} \frac{1}{k!} \sum_{\alpha\in[d]^k} \boldsymbol{x}^\alpha \cdot \partial_\alpha f(\boldsymbol{0}). \tag{5.4}$$

5.3.3. Definition (Smooth function). We say a function $f : \mathbb{R}^d \to \mathbb{R}$ is smooth with constant $c > 0$ if it satisfies the following: f is analytic and for every $k \in \mathbb{N}$, $\boldsymbol{x} \in \mathbb{R}^d$ and $\alpha \in [d]^k$, we have

$$|\partial_\alpha f(\boldsymbol{x})| \leq c^k k^{\frac{k}{2}}. \tag{5.5}$$

Oracle access. As mentioned earlier, many quantum optimization procedures assuming access to the objective function p via a probability oracle, defined as follows

$$U_p : |\boldsymbol{x}\rangle|0\rangle \to \sqrt{p(\boldsymbol{x})}|\boldsymbol{x}\rangle|1\rangle + \sqrt{1-p(\boldsymbol{x})}|\boldsymbol{x}\rangle|0\rangle \quad \text{for every } \boldsymbol{x}, \tag{5.6}$$

where the continuous input variable $\boldsymbol{x}$ is represented as a finite-precision binary encoding of $\boldsymbol{x}$. However, for most of the quantum techniques that we employ to improve the gradient-descent algorithm, it is more natural to work with a *phase oracle*, acting as

$$\mathrm{O}_p : |\boldsymbol{x}\rangle \to e^{ip(\boldsymbol{x})}|\boldsymbol{x}\rangle \quad \text{for every } \boldsymbol{x}. \tag{5.7}$$

For technical reasons we assume that we can perform fractional queries as well, defined as follows: for $r \in [-1, 1]$, the action of the *fractional oracle* O_{rp} is given by

$$\mathrm{O}_{rp} : |x\rangle|\vec{0}\rangle \to e^{irp(x)}|x\rangle|\vec{0}\rangle \quad \text{for every } \boldsymbol{x}.$$

Using the Linear Combination of Unitaries (LCU) techniques [BCC+15], we show [GAW17] how to efficiently simulate a phase oracle with precision ε, using $O(\log(1/\varepsilon))$ queries to the probability oracle U_p. Similarly, we show that under some reasonable conditions, we can simulate the probability oracle U_p with ε precision, using $O(\log(1/\varepsilon))$ queries to the phase oracle O_p. Finally, we show an efficient inter-conversion between the probability oracle and fractional query oracle. For the purposes of our chapter the efficient simulation, between probability, phase and fractional query oracles, essentially means that we can interchangeably work with any of these oracles using whichever fits our setting best. We are not aware of any prior result that shows this simulation and we believe that our oracle conversion techniques could be useful for other applications.

5.3.2 Overview of Jordan's algorithm

Sketch of the algorithm. Stephen Jordan constructed a surprisingly simple quantum algorithm [Jor05, Bul05b] that can approximately calculate the d-dimensional gradient of a function $f : \mathbb{R}^d \to \mathbb{R}$ with a *single* evaluation of f. In contrast, using standard classical techniques, one would use $d + 1$ function evaluations to calculate the gradient at a point $\boldsymbol{x} \in \mathbb{R}^d$: one can first evaluate $f(\boldsymbol{x})$ and then, for every $i \in [d]$, evaluate $f(\boldsymbol{x} + \delta \boldsymbol{e}_i)$ (for some $\delta > 0$) to get an approximation of the gradient in direction i using the standard formula

$$\nabla_i f(\boldsymbol{x}) \approx \frac{f(\boldsymbol{x} + \delta \boldsymbol{e}_i) - f(\boldsymbol{x})}{\delta}.$$

The basic idea of Jordan's quantum algorithm [Jor05] is simple and uses the following two observations. If f is twice differentiable at $\boldsymbol{x}$, then using Taylor's theorem for multivariate functions, we have $f(\boldsymbol{x} + \boldsymbol{\delta}) = f(\boldsymbol{x}) + \nabla f \cdot \boldsymbol{\delta} + O(\|\boldsymbol{\delta}\|^2)$, which in particular implies that for small $\|\boldsymbol{\delta}\|$, the function f is very close to being affine linear. The second observation is that, using the value of $f(\boldsymbol{x} + \boldsymbol{\delta})$, one can implement a fractional query oracle:

$$\mathrm{O}_{2\pi Sf} : |\boldsymbol{\delta}\rangle \to e^{2\pi i S f(\boldsymbol{x}+\boldsymbol{\delta})}|\boldsymbol{\delta}\rangle \approx e^{2\pi i S f(\boldsymbol{x})} e^{2\pi i S \nabla f \cdot \boldsymbol{\delta}}|\boldsymbol{\delta}\rangle, \tag{5.8}$$

where the approximation uses $f(\boldsymbol{x} + \boldsymbol{\delta}) \approx f(\boldsymbol{x}) + \nabla f \cdot \boldsymbol{\delta}$ for small $\|\boldsymbol{\delta}\|$. The role of the scaling factor S is to make make the phases appropriate for the final quantum Fourier transform.

We now describe Jordan's algorithm. Assume that all real vectors are expressed up to some finite amount of precision. In order to compute the gradient at $\boldsymbol{x}$, let $G_{\boldsymbol{x}}^d$ be a sufficiently small discretized d-dimensional grid around $\boldsymbol{x}$. The

algorithm starts with a uniform superposition $|\psi\rangle = \frac{1}{\sqrt{|G_{\boldsymbol{x}}^d|}} \sum_{\boldsymbol{\delta} \in G_{\boldsymbol{x}}^d} |\boldsymbol{\delta}\rangle$ and applies the phase oracle $\mathrm{O}_{2\pi Sf}$ (in Eq. (5.8)) to $|\psi\rangle$. Next, the following linear map (which is unitarily equivalent to the quantum Fourier transform)

$$\mathrm{QFT}_{G_{\boldsymbol{x}}^d} : |x\rangle \to \frac{1}{\sqrt{G_{\boldsymbol{x}}^d}} \sum_{k \in G_{\boldsymbol{x}}^d} e^{-2\pi i |G_{\boldsymbol{x}}^d| xk} |k\rangle$$

is applied to the resulting state and each register is measured to obtain the gradient of f at $\boldsymbol{x}$ approximately. Due to approximate linearity of the phase (see Eq. (5.8)), observe that the $\mathrm{QFT}_{G_{\boldsymbol{x}}^d}$ map will approximately give us the gradient. This algorithm uses $\mathrm{O}_{2\pi Sf}$ once and Jordan showed how to implement $\mathrm{O}_{2\pi Sf}$ using one sufficiently precise binary oracle evaluation.

Complexity of the algorithm. It remains to pick the parameters of the grid and the constant S in Eq. (5.8). For simplicity, assume that $\|\nabla f(\boldsymbol{x})\|_\infty \leq 1$, and suppose we want to approximate $\nabla f(\boldsymbol{x})$ ε-coordinate-wise accuracy, with high success probability. Under the assumption that "the 2nd partial derivatives of f have a magnitude of approximately D_2", Jordan argues[5] that choosing $G_{\boldsymbol{x}}^d$ to be a d-dimensional hypercube with edge length $\ell \approx \frac{\varepsilon}{D_2\sqrt{d}}$ and with $N \approx \frac{1}{\varepsilon}$ equally spaced grid points in each dimension, the quantum algorithm yields an ε-approximate gradient by setting $S = \frac{N}{\ell} \approx \frac{D_2\sqrt{d}}{\varepsilon^2}$. Moreover, since the Fourier transform is relatively insensitive to local phase errors it suffices to implement the phase $Sf(\boldsymbol{x} + \boldsymbol{\delta})$ up to some constant, say 1% accuracy.

Our improvements. We improve on the results of Jordan [Jor05] in a couple of ways.

1. We first remark that the analysis of the quantum algorithm presented by Jordan [Jor05] is not complete. During the derivation of the above parameters, Jordan makes the assumption that the third and higher-order terms of the Taylor expansion of p around $\boldsymbol{x}$ are negligible, however it is not clear from his work [Jor05] how to actually handle the case when they are non-negligible. This could be a cause of concern for the runtime analysis, since these higher-order terms potentially introduce a dependence on the dimension d, which was indeed the case when we rigorously analyzed Jordan's algorithm using the probability oracle.

2. As discussed in Section 5.2.2, we realized that in applications of the gradient-descent algorithm for optimization problems, instead of the η-accurate binary oracle, it is natural to assume access to the probability oracle, which we

[5] We specifically refer to equation (4) in [Jor05] (equation (3) in the arXiv version), and the discussion afterwards. Note that our precision parameter ε corresponds to the uncertainty parameter σ in [Jor05].

showed to be equivalent to the phase oracle $\mathrm{O}_f : |\boldsymbol{x}\rangle \to e^{ip(\boldsymbol{x})}|\boldsymbol{x}\rangle$ (and fractional query oracles). In order to use Jordan's original algorithm to obtain the gradient with ε-accuracy, one needs to implement the query oracle O_f^S for the parameter $S \approx \frac{D_2\sqrt{d}}{\varepsilon^2}$, which can be achieved using $\lceil S \rceil$ consecutive (fractional) queries. Although this would give a square-root dependence on d it scales as $O(1/\varepsilon^2)$ with the precision.

In this work, we improve the quadratic dependence on $1/\varepsilon$ to essentially linear. Additionally, we *rigorously* prove the square-root scaling with d under reasonable assumptions on the derivatives of p, which was absent in the prior work of Jordan [Jor05]. We describe the algorithm in the next section, but first present our main result, whose proof is deferred to the end of this section.

5.3.4. Theorem. *Suppose $p : \mathbb{R}^d \to \mathbb{R}$ is smooth (as in Definition 5.3.3) with constant $c > 0$. Let $\varepsilon \leq c$. Then, there is a quantum algorithm that, on input $\boldsymbol{x} \in \mathbb{R}^d$, outputs an ε-approximate gradient $\tilde{\nabla} p(\boldsymbol{x}) \in \mathbb{R}^d$ such that*

$$\|\nabla p(\boldsymbol{x}) - \tilde{\nabla} p(\boldsymbol{x})\|_\infty \leq \varepsilon,$$

with probability at least $1-\delta$, using $\widetilde{O}(\frac{c\sqrt{d}}{\varepsilon} \log\left(\frac{1}{\delta}\right))$ queries to the phase oracle O_p.

Note that in this theorem, just as in the rest of this chapter, we assume that f is analytic on $\mathbb{R}^d$ rather than on a compact domain of $\mathbb{R}^d$. This assumption is not necessary but makes the statements simpler. It is straightforward to translate the results when the function is only defined on a subset of $\mathbb{R}^d$. However, a finite domain imposes restrictions to the evaluation points of the function.

5.3.3 Rigorous analysis of Jordan's algorithm

We now describe Jordan's algorithm in more detail and provide a generic analysis of its behavior. In the next subsection we improve the results presented here using our finite difference methods. Before describing the algorithm, we introduce appropriate representation of our qubit strings suitable for fixed-point arithmetics.

5.3.5. Definition. For every $b \in \{0,1\}^n$, let $j^{(b)} \in \{0, \ldots, 2^n - 1\}$ be the integer corresponding to the binary string $b = (b_1, \ldots, b_n)$. We label the n-qubit basis state $|b_1\rangle|b_2\rangle \cdots |b_n\rangle$ by $|x^{(b)}\rangle$, where

$$x^{(b)} = \frac{j^{(b)}}{2^n} - \frac{1}{2} + 2^{-n-1}.$$

We denote the set of corresponding labels as

$$G_n := \left\{ \frac{j^{(b)}}{2^n} - \frac{1}{2} + 2^{-n-1} : j^{(b)} \in \{0, \ldots, 2^n - 1\} \right\}.$$

Note that there is a bijection between $\{j^{(b)}\}_{b\in\{0,1\}^n}$ and $\{x^{(b)}\}_{b\in\{0,1\}^n}$, so we will use $|x^{(b)}\rangle$ and $|j^{(b)}\rangle$ interchangeably. In the rest of this section we always label n-qubit basis states by elements of G_n.

5.3.6. Definition. For $x \in G_n$ we define the state $|x\rangle$ as follows

$$\mathrm{QFT}_{G_n} : |x\rangle \to \frac{1}{\sqrt{2^n}} \sum_{k\in G_n} e^{2\pi i 2^n xk} |k\rangle.$$

5.3.7. Claim. *This unitary* QFT_{G_n} *is the same as the usual quantum Fourier transform up to conjugation with a tensor product of* n *single-qubit unitaries.*

Proof. For bit strings $b, c \in \{0,1\}^n$, let $x^{(b)} \in G_n$ and $j^{(b)} \in \{0,\dots,2^n-1\}$, be as defined in Definition 5.3.5. Then QFT_{G_n} acts on $|j^{(b)}\rangle \equiv |x^{(b)}\rangle$ as

$$\begin{aligned}
\mathrm{QFT}_{G_n} &: |x^{(b)}\rangle \\
&\to \frac{1}{\sqrt{2^n}} \sum_{x^{(c)}\in G_n} e^{2\pi i 2^n x^{(b)} x^{(c)}} |x^{(c)}\rangle \\
&\equiv \frac{1}{\sqrt{2^n}} \sum_{j^{(c)}\in\{0,\dots,2^n-1\}} e^{2\pi i 2^n \left(\frac{j^{(b)}}{2^n} - \frac{1}{2} + 2^{-n-1}\right)\left(\frac{j^{(c)}}{2^n} - \frac{1}{2} + 2^{-n-1}\right)} |j^{(c)}\rangle \\
&\equiv \frac{1}{\sqrt{2^n}} \sum_{j^{(c)}\in\{0,\dots,2^n-1\}} e^{2\pi i \left(\frac{j^{(b)} j^{(c)}}{2^n} - \left(j^{(b)} + j^{(c)}\right)\left(\frac{1}{2} + 2^{-n-1}\right) + \left(2^{n-2} - \frac{1}{2} + 2^{-n-2}\right)\right)} |j^{(c)}\rangle.
\end{aligned}$$

Using the usual quantum Fourier transform

$$\mathrm{QFT}_n : |j^{(b)}\rangle \to \frac{1}{\sqrt{2^n}} \sum_{j^{(c)}\in\{0,\dots,2^n-1\}} e^{2\pi i 2^{-n} j^{(b)} j^{(c)}} |j^{(c)}\rangle$$

and the phase unitary

$$U : |j^{(b)}\rangle \to e^{2\pi i \left(-j^{(b)}\left(\frac{1}{2} + 2^{-n-1}\right) + \left(2^{n-2} - \frac{1}{2} + 2^{-n-2}\right)/2\right)} |j^{(b)}\rangle \quad \text{for } j^{(b)} \in \{0,\dots,2^n-1\},$$

observe that

$$\mathrm{QFT}_{G_n} = U \cdot \mathrm{QFT}_n \cdot U.$$

Writing $j^{(b)}$ in binary, we can express U as a tensor product of n phase gates. □

Now we are ready to describe Jordan's quantum gradient calculation algorithm and give a rigorous analysis of it.

Algorithm 1 Jordan's quantum gradient calculation algorithm

Registers: Use n-qubit input registers $|x_1\rangle|x_2\rangle\cdots|x_d\rangle$ with each qubit set to $|0\rangle$.

Labels: Label the n-qubit states of each register with elements of G_n as in Definition 5.3.5.

Input: A function $f : G_n^d \to \mathbb{R}$ with phase-oracle O_f access such that

$$\mathrm{O}_f^{\pi 2^{n+1}}|x_1\rangle|x_2\rangle\cdots|x_d\rangle = e^{2\pi i 2^n f(x_1,x_2,\ldots,x_d)}|x_1\rangle|x_2\rangle\cdots|x_d\rangle.$$

1: **Init** Apply a Hadamard transform to each qubit of the input registers.
2: **Oracle call** Apply the modified phase oracle $\mathrm{O}_f^{\pi 2^{n+1}}$ on the input registers.
3: $\mathbf{QFT}_{G_n}^{-1}$ Fourier transform each register individually:

$$|x\rangle \to \frac{1}{\sqrt{2^n}} \sum_{k\in G_n} e^{-2\pi i 2^n xk}|k\rangle.$$

4: **Measure** each input register, denote the measurement outcome from the jth register by k_j.
5: **Output** $(k_1, k_2, \ldots, k_d)$ as the estimation for the gradient.

5.3.8. Lemma. *Let $N = 2^n$, $c \in \mathbb{R}$ and $\boldsymbol{g} \in \mathbb{R}^d$ such that $\|\boldsymbol{g}\|_\infty \leq 1/3$. If $f : G_n^d \to \mathbb{R}$ is such that*

$$|f(\boldsymbol{x}) - \boldsymbol{g}\cdot\boldsymbol{x} - c| \leq \frac{1}{42\pi N}, \tag{5.9}$$

for all but a $1/1000$ fraction of the points $x \in G_n^d$, then the output of Algorithm 1 satisfies:

$$\Pr\left[|k_i - g_i| > 4/N\right] \leq 1/3 \quad \textit{for every } i \in [d].$$

Proof. First, note that $|G_n| = N$ from Definition 5.3.5. Consider the following quantum states

$$|\phi\rangle := \frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{x}\in G_n^d} e^{2\pi i N f(\boldsymbol{x})}|\boldsymbol{x}\rangle \qquad \text{and} \qquad |\psi\rangle := \frac{1}{\sqrt{N^d}} \sum_{\boldsymbol{x}\in G_n^d} e^{2\pi i N(\boldsymbol{g}\cdot\boldsymbol{x}+c)}|\boldsymbol{x}\rangle.$$

Note that $|\phi\rangle$ is the state we obtain in Algorithm 1 after line 2 and $|\psi\rangle$ is its "ideal version" that we try to approximate with $|\phi\rangle$. Observe that the "ideal" $|\psi\rangle$ is actually a product state:

$$|\psi\rangle = e^{2\pi i N c}\Big(\frac{1}{\sqrt{N}} \sum_{x_1\in G_n} e^{2\pi i N g_1\cdot x_1}|x_1\rangle\Big) \otimes \cdots \otimes \Big(\frac{1}{\sqrt{N}} \sum_{x_d\in G_n} e^{2\pi i N g_d\cdot x_d}|x_d\rangle\Big).$$

It is easy to see that after applying the inverse Fourier transform to each register separately (as in line 3) to $|\psi\rangle$, we obtain the state

$$\Big(\frac{1}{N}\sum_{(x_1,k_1)\in G_n^2} e^{2\pi i N x_1(g_1-k_1)}|k_1\rangle\Big)\otimes\cdots\otimes\Big(\frac{1}{N}\sum_{(x_d,k_d)\in G_n^2} e^{2\pi i N x_d(g_d-k_d)}|k_d\rangle\Big).$$

Suppose we make a measurement and observe $(k_1,\ldots,k_d)$. As shown in the analysis of phase estimation [NC02], we have the following[6]: for every $i\in[d]$ (for a fixed accuracy parameter $\kappa>1$), the following holds:

$$\Pr\Big[|k_i-\nabla_i f|>\frac{\kappa}{N}\Big]\leq\frac{1}{2(\kappa-1)}.$$

By fixing $\kappa=4$, we obtain the desired conclusion of the theorem, i.e., if we had access to $|\psi\rangle$ (instead of $|\phi\rangle$), then for each coordinate, we would get a $4/N$-approximation of the gradient with probability at least $5/6$. It remains to show that this probability does not change more than $1/3-1/6=1/6$ if we apply the Fourier transform to $|\phi\rangle$ instead of $|\psi\rangle$. Observe that the difference in the probability of any measurement outcome on these states is bounded by twice the trace distance between $|\psi\rangle$ and $|\phi\rangle$,

$$\||\psi\rangle\langle\psi|-|\phi\rangle\langle\phi|\|_{\mathrm{Tr}}=2\sqrt{1-|\langle\psi|\phi\rangle|^2}\leq 2\||\psi\rangle-|\phi\rangle\|. \tag{5.10}$$

Since the Fourier transform is unitary and does not change the Euclidean distance, it suffices to show $\||\psi\rangle-|\phi\rangle\|\leq 1/12$ in order to conclude the theorem. Let $S\subseteq G_n^d$ denote the set of points satisfying Eq. (5.9).

$$\begin{aligned}
&\||\phi\rangle-|\psi\rangle\|^2\\
&=\frac{1}{N^d}\sum_{\boldsymbol{x}\in G_n^d}\left|e^{2\pi i N f(\boldsymbol{x})}-e^{2\pi i N(\boldsymbol{g}\cdot\boldsymbol{x}+c)}\right|^2\\
&=\frac{1}{N^d}\sum_{\boldsymbol{x}\in S}\left|e^{2\pi i N f(\boldsymbol{x})}-e^{2\pi i N(\boldsymbol{g}\cdot\boldsymbol{x}+c)}\right|^2+\frac{1}{N^d}\sum_{\boldsymbol{x}\in G_n^d\setminus S}\left|e^{2\pi i N f(\boldsymbol{x})}-e^{2\pi i N(\boldsymbol{g}\cdot\boldsymbol{x}+c)}\right|^2\\
&\leq\frac{1}{N^d}\sum_{\boldsymbol{x}\in S}|2\pi N f(\boldsymbol{x})-2\pi N(\boldsymbol{g}\cdot\boldsymbol{x}+c)|^2+\frac{1}{N^d}\sum_{\boldsymbol{x}\in G_n^d\setminus S}4\quad(\text{using }|e^{iz}-e^{iy}|\leq|z-y|)\\
&=\frac{1}{N^d}\sum_{\boldsymbol{x}\in S}(2\pi N)^2\,|f(\boldsymbol{x})-(\boldsymbol{g}\cdot\boldsymbol{x}+c)|^2+4\frac{|G_n^d\setminus S|}{N^d}\\
&\leq\frac{1}{N^d}\sum_{\boldsymbol{x}\in S}\left(\frac{1}{21}\right)^2+\frac{4}{1000}\qquad\qquad(\text{by the assumptions of the theorem})\\
&\leq\frac{1}{441}+\frac{1}{250}<\frac{1}{144}=\left(\frac{1}{12}\right)^2.
\end{aligned}$$

□

[6]Note that our Fourier transform is slightly altered, but the same proof applies as in [NC02, Eq. (5.34)]. In fact this result can be directly translated to our case by considering the unitary conjugations proven in Remark 5.3.7.

In the following theorem we assume that we have access to (a high power of) a phase oracle of a function f that is very well approximated by an affine linear function $\boldsymbol{g} \cdot \boldsymbol{z} + c$ on a hypergrid with edge-length $r \in \mathbb{R}$ around some $\boldsymbol{y} \in \mathbb{R}^d$. These assumptions were made informally by Jordan [Jor05] as well. We show that if the relative precision of the approximation is precise enough, then Algorithm 1 can compute an approximation of the gradient $\boldsymbol{g}$ with small query and gate complexity.

5.3.9. THEOREM. *Let $c \in \mathbb{R}$ and $r, \delta, \varepsilon \leq 1$. Fix $\boldsymbol{y} \in \mathbb{R}^d$. Let $n_\varepsilon = \lceil \log(4/(r\varepsilon)) \rceil$, $n_1 = \lceil \log(3r) \rceil$ and $n = n_\varepsilon + n_1$. Suppose $\|g\|_\infty \leq 1$ and $f : \left(\boldsymbol{y} + rG_n^d\right) \to \mathbb{R}$ satisfies*

$$|f(\boldsymbol{y} + r\boldsymbol{x}) - \boldsymbol{g} \cdot r\boldsymbol{x} - c| \leq \frac{\varepsilon r}{8 \cdot 42\pi} \tag{5.11}$$

for all but a $1/1000$ fraction of the points $\boldsymbol{x} \in G_n^d$. Then, given access to a phase oracle $O'_f : |\boldsymbol{x}\rangle \to e^{2\pi i 2^{n_\varepsilon} f(\boldsymbol{y}+r\boldsymbol{x})}|\boldsymbol{x}\rangle$ acting on $\mathcal{H} = \mathrm{Span}\{|\boldsymbol{x}\rangle : \boldsymbol{x} \in G_n^d\}$, Algorithm 1 outputs a vector $\tilde{\boldsymbol{g}} \in \mathbb{R}^d$ such that

$$\Pr\left[\, \|\tilde{\boldsymbol{g}} - \boldsymbol{g}\|_\infty \leq \varepsilon \right] \geq 1 - \delta,$$

using $O(\log\left(\frac{d}{\delta}\right))$ queries to O'_f and $\widetilde{O}\Big(d \log\left(\frac{d}{\delta}\right) \log\left(1/\varepsilon\right)\Big)$ other gates.

Proof. Let $N_1 := 2^{n_1}$, $N := 2^n$, and $h(\boldsymbol{x}) := \frac{f(\boldsymbol{y}+r\boldsymbol{x})}{N_1}$, then using Eq. (5.11), it follows that

$$\Big|h(x) - \frac{\boldsymbol{g} \cdot r\boldsymbol{x}}{N_1} - \frac{c}{N_1}\Big| \leq \frac{\varepsilon r}{8 \cdot 42\pi N_1} \leq \frac{1}{42\pi N}.$$

Note that $\mathrm{O}'_f = \mathrm{O}_h^{2\pi N}$. Using Lemma 5.3.8, it follows that, with probability at least $2/3$, Algorithm 1 outputs a $\tilde{\boldsymbol{g}}$ such that, for every $i \in [d]$, we have $\left|\tilde{g}_i - \frac{r}{N_1} g_i\right| \leq \frac{4}{N}$ with probability at least $2/3$. In particular, we have that

$$\Big|\frac{N_1}{r}\tilde{g}_i - g_i\Big| \leq \frac{4N_1}{rN} = \frac{4}{rN_\varepsilon} \leq \varepsilon.$$

By repeating the procedure $O(\log(d/\delta))$ times and taking the median coordinate-wise, we get a vector $\tilde{\boldsymbol{g}}_{\mathrm{med}}$, such that, with probability at least $(1 - \delta)$, we have $\|\tilde{\boldsymbol{g}}_{\mathrm{med}} - \boldsymbol{g}\|_\infty \leq \varepsilon$.

The gate complexity statement follows from the fact that the complexity of Algorithm 1 is dominated by that of the d independent quantum Fourier transforms, each of which can be approximately implemented using $O(n \log n)$ gates (see [Wol13, Section 4.5]). We repeat the procedure $O(\log(d/\delta))$ times, which amounts to $O(d \log(d/\delta) n \log n)$ gates. The final median computation can be done in $\widetilde{O}(\log(d/\delta) n \log n)$ gates overall. So the final gate complexity is given by $O(d \log(d/\delta) n \log n)$, which gives the stated gate complexity by observing that $n = O(\log(1/\varepsilon))$. □

5.3.4 Improved quantum gradient algorithm using higher-degree methods

Theorem 5.3.9 shows that Jordan's algorithm works well if the function is very close to a linear function over a large hypercube. However, in general even highly regular functions tend to quickly diverge from their linear approximations. To tackle this problem we borrow ideas from numerical analysis and use higher-degree finite-difference formulas to extend the range of approximate linearity.

Now we describe the general central-difference approximation formulas, which are the basis for our improvements. Central-difference formulas (see e.g., [Li05]) are often used to give precise approximations of derivatives of a function $f : \mathbb{R}^d \to \mathbb{R}$.[7] These formulas yield precise approximations of directional derivatives too, and thus we use them to approximate the gradient of a high-dimensional function.

5.3.10. Definition. Let $m \geq 1$. The degree-$(2m)$ *central-difference approximation* of a function $f : \mathbb{R}^d \to \mathbb{R}$ is:

$$f_{(2m)}(\boldsymbol{x}) := \sum_{\substack{\ell=-m \\ \ell\neq 0}}^{m} \frac{(-1)^{\ell-1}}{\ell} \frac{\binom{m}{|\ell|}}{\binom{m+|\ell|}{|\ell|}} f(\ell \boldsymbol{x}). \tag{5.12}$$

The corresponding *central-difference coefficients* for $\ell \in \{-m, \ldots, m\} \backslash \{0\}$ are given by

$$a_\ell^{(2m)} := \frac{(-1)^{\ell-1}}{\ell} \frac{\binom{m}{|\ell|}}{\binom{m+|\ell|}{|\ell|}} \quad \text{and } a_0^{(2m)} := 0.$$

Using this definition, we prove the following lemma, which gives bounds on the coefficients $a_\ell^{(2m)}$ in the central-difference formulas. A proof of this lemma can be found in [GAW17, Appendix A].

5.3.11. Lemma. *Suppose $m \in \mathbb{N}$ and $f : \mathbb{R} \to \mathbb{R}$ is $(2m+1)$-times differentiable. Then for all $\delta \in \mathbb{R}_+$*

$$\left|f'(0)\delta - f_{(2m)}(\delta)\right| = \left|f'(0)\delta - \sum_{\ell=-m}^{m} a_\ell^{(2m)} f(\ell\delta)\right| \leq e^{-\frac{m}{2}} \left\|f^{(2m+1)}\right\|_\infty |\delta|^{2m+1}, \tag{5.13}$$

where $\left\|f^{(2m+1)}\right\|_\infty := \sup_{\xi\in[-\ell\delta,\ell\delta]} |f^{(2m+1)}(\xi)|$ and $a_\ell^{(2m)}$ coefficients are defined as in Definition 5.3.10. Moreover

$$\sum_{\ell=-m}^{m} \left|a_\ell^{(2m)}\right| < 2\sum_{\ell=1}^{m} \frac{1}{\ell} \leq 2\ln m + 2. \tag{5.14}$$

[7] There are a variety of other related formulas [Li05], but we stick to the central-difference because the absolute values of the coefficients using this formula scale favourably with the approximation degree.

This lemma shows that if a function $f : \mathbb{R} \to \mathbb{R}$ is $(2m+1)$-times continuously differentiable, then for small enough δ the approximation error in (5.12) is upper bounded by a factor proportional to δ^{2m+1}. If $\left\|f^{(2m+1)}\right\|_\infty \leq c^m$ for all m and we choose $\delta < 1/c$, then the approximation error becomes exponentially small in m, motivating the use of higher-degree methods in our modified gradient calculation algorithm. In [GAW17, Appendix A], we generalized Lemma 5.3.11 to higher-degree functions $f : \mathbb{R}^d \to \mathbb{R}$ and proved the following result about analytic functions. This theorem essentially shows that the right hand side of Eq. (5.12) is a good approximation to $\nabla f(\mathbf{0})\boldsymbol{y}$ (where the approximation factor depends on m, d).

5.3.12. Theorem. *Let $R > 0$. Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is smooth with constant $c > 0$. Then,*

$$|\nabla f(\mathbf{0})\boldsymbol{y} - f_{(2m)}(\boldsymbol{y})| \leq \sum_{k=2m+1}^{\infty} \left(8Rcm\sqrt{d}\right)^k,$$

for all but a $1/1000$ fraction of points $\boldsymbol{y} \in R \cdot G_n^d$.

We are now ready to use this result and prove our main theorem. Our main gradient calculation algorithm is similar to Jordan's orginal algorithm. Instead of the binary oracle, we assume access to the fractional query oracle and we apply Jordan's algorithm (i.e., Algorithm 1) to the finite difference approximation $f_{(2m)}$ of the gradient instead of the function f itself. Under reasonable smoothness assumptions, we show that the complexity of Algorithm 1 when applied to functions evaluated using a central-difference formula, is $\widetilde{O}(\sqrt{d}/\varepsilon)$. This gives us our main theorem, which we restate below for convenience.

5.3.13. Theorem. *Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is smooth with constant $c > 0$. Fix $\varepsilon \leq c$. Then, there is a quantum algorithm that, on input $\boldsymbol{x} \in \mathbb{R}^d$, outputs an ε-approximate gradient $\widetilde{\nabla} f(\boldsymbol{x}) \in \mathbb{R}^d$ such that*

$$\|\nabla f(\boldsymbol{x}) - \widetilde{\nabla} f(\boldsymbol{x})\|_\infty \leq \varepsilon,$$

with probability at least $1-\delta$, using $\widetilde{O}(\frac{c\sqrt{d}}{\varepsilon} \log\left(\frac{1}{\delta}\right))$ queries to the phase oracle O_f.

Proof. Let $g(\boldsymbol{y}) := f(\boldsymbol{x} + \boldsymbol{y})$ for every $y \in \mathbb{R}^d$. By Theorem 5.3.12 we know that for a uniformly random $\boldsymbol{y} \in R \cdot G_n^d$ we have $|\nabla g(\mathbf{0})\boldsymbol{y} - g_{(2m)}(\boldsymbol{y})| \leq \sum_{k=2m+1}^{\infty} \left(8Rcm\sqrt{d}\right)^k$ with probability at least $999/1000$. Now we choose R such that this infinite summation becomes smaller that $\frac{\varepsilon R}{8 \cdot 42\pi}$. For that, let

$R^{-1} = 8cm\sqrt{d}\Big(81 \cdot 4 \cdot 42\pi cm\sqrt{d}/\varepsilon\Big)^{1/(2m)}$, then

$$\begin{aligned}
\sum_{k=2m+1}^{\infty} \Big(8Rcm\sqrt{d}\Big)^k &= \Big(8Rcm\sqrt{d}\Big)^{2m+1} \sum_{k=0}^{\infty} \Big(8Rcm\sqrt{d}\Big)^k \\
&\leq \frac{\varepsilon}{81 \cdot 4 \cdot 42\pi cm\sqrt{d}} \Big(81 \cdot 4 \cdot 42\pi cm\sqrt{d}/\varepsilon\Big)^{\frac{-1}{2m}} \sum_{k=0}^{\infty} \Big(\frac{8}{9}\Big)^k \\
&\qquad \text{(by our choice of } R\text{)} \\
&\leq \frac{\varepsilon}{8cm\sqrt{d} \cdot 4 \cdot 42\pi} \Big(81 \cdot 4 \cdot 42\pi cm\sqrt{d}/\varepsilon\Big)^{\frac{-1}{2m}} \\
&\qquad \text{(since } \textstyle\sum_{k=0}^{\infty} \big(\frac{8}{9}\big)^k \leq 9\text{)} \\
&= \frac{\varepsilon R}{4 \cdot 42\pi}.
\end{aligned}$$

Using Theorem 5.3.9, we can compute an approximate gradient with $O(\log(d/\delta))$ queries to $\mathrm{O}^S_{g_{(2m)}}$, where $S = O(\frac{1}{\varepsilon R})$. Observe that

$$\mathrm{O}^S_{g_{(2m)}}|\boldsymbol{y}\rangle = e^{iSg_{(2m)}(\boldsymbol{y})}|\boldsymbol{y}\rangle = e^{iS\sum_{\ell=-m}^{m} a_\ell^{(2m)} g(\ell \boldsymbol{y})}|\boldsymbol{y}\rangle.$$

Using the relation $g(\boldsymbol{y}) = f(\boldsymbol{x}+\boldsymbol{y})$, it is easy to see that the number of phase queries to O_f to implement a modified oracle call $\mathrm{O}^S_{g_{(2m)}}$ is

$$\sum_{\ell=-m}^{m} \left\lceil \left| a_\ell^{(2m)} \right| S \right\rceil \leq 2m + S \sum_{\ell=-m}^{m} a_\ell^{(2m)} \leq 2m + S\,(2\ln m + 2)\,, \tag{5.15}$$

where the second inequality used Eq. (5.14). Then $\mathrm{O}^S_{g_{(2m)}}$ can be implemented using $O(\frac{\log m}{\varepsilon R} + m)$ fractional queries to O_f. By choosing $m = \log(c\sqrt{d}/\varepsilon)$ the query complexity becomes[8]

$$O\Big(\frac{\log m}{\varepsilon R} + m\Big) = O\Big(\frac{c\sqrt{d}}{\varepsilon} m \log m\Big) = O\Big(\frac{c\sqrt{d}}{\varepsilon} \log\Big(\frac{c\sqrt{d}}{\varepsilon}\Big) \log\log\Big(\frac{c\sqrt{d}}{\varepsilon}\Big)\Big). \tag{5.16}$$

□

The above achieves, up to logarithmic factors, the desired $1/\varepsilon$ scaling in the precision parameter and also the $\sqrt{d}$ scaling with the dimension. This improves the results of [Jor05] both quantitatively and qualitatively.

[8] We remark that if we strengthen the $c^k k^{\frac{k}{2}}$ upper bound assumption on the derivatives to c^k, then we could improve the bound of Theorem 5.3.12 by a factor of $k^{-k/2}$. Therefore in the definition of R^{-1} we could replace m by $\sqrt{m}$ which would quadratically improve the log factor in (5.16).

5.4 Other results

In [GAW17], there were many other results which we did not include in this chapter. In this section, we briefly summarize those results and give proof sketches of some results there.

5.4.1 Smoothness of probability oracles

We show that the seemingly strong requirement of Theorem 5.3.4 is naturally satisfied by probability oracles arising from typical quantum optimization protocols. In such protocols, probability oracles usually correspond to the measurement outcome probability of some orthogonal projector Π on the output state of a parametrized circuit $U(\boldsymbol{x})$ acting on some fixed initial state $|\psi\rangle$, i.e.,

$$p(\boldsymbol{x}) = \langle\psi|U(\boldsymbol{x})^\dagger \Pi U(\boldsymbol{x})|\psi\rangle.$$

Usually the parametrized circuit can be written as

$$U(\boldsymbol{x}) = U_0 \prod_{j=1}^{d} \left(e^{ix_j H_j}\right) U_j,$$

where the U_js are fixed unitaries and the H_js are fixed Hermitian operators. We can assume without loss of generality that $\|H_j\| \leq 1/2$. Under these conditions we can show that p is analytic, and all partial derivatives of p are upper bounded by 1 in magnitude, i.e., p is smooth and satisfies the conditions of Theorem 5.3.4. For more details, see [GAW17, Lemma 25-26].

5.4.2 Lower bounds for gradient computation

An interesting question is whether we can improve the classical $O(d/\varepsilon^2)$-gradient computation algorithm by a super-quadratic factor? At first sight it would very well seem possible considering that our algorithm gains a speed-up using the quantum Fourier transform. However, we show that in general this is *not* possible for smooth non-polynomial functions, and give a lower bound of $\Omega(\sqrt{d}/\varepsilon)$ for the complexity of a generic quantum gradient calculation algorithm. The precise statement can be found in [GAW17, Theorem 25], and below we state the theorem informally.

5.4.1. Theorem (Informal). *Let $\varepsilon, d > 0$. There exists a family of smooth functions $\mathcal{F} \subseteq \{f : \mathbb{R}^d \to \mathbb{R}\}$ such that the following holds. Every quantum algorithm $\mathcal{A}$ that makes T queries to the phase oracle O_f and for every $f \in \mathcal{F}$, $\mathcal{A}$ outputs, with probability $\geq 2/3$, an approximate gradient $\boldsymbol{g} \in \mathbb{R}^d$ satisfying*

$$\|\boldsymbol{g} - \nabla f(\boldsymbol{0})\|_\infty < \varepsilon,$$

needs to make $T = \Omega(\sqrt{d}/\varepsilon)$ queries.

Proof sketch. We exhibit a family of functions $\mathcal{F}$ for which the corresponding phase oracles $\{O_f : f \in \mathcal{F}\}$ are hard to distinguish form each other, but the functions in $\mathcal{F}$ can be uniquely identified by calculating their gradient at $\mathbf{0}$ with accuracy ε. In particular, this implies that calculating an approximation of the gradient vector for these functions must be at least as hard as distinguishing the phase oracles corresponding to functions in $\mathcal{F}$. At this point, we use the so-called hybrid method [BBBV97] to show that distinguishing the phase oracles O_f corresponding to functions $f \in \mathcal{F}$ requires $\Omega(\sqrt{d}/\varepsilon)$ queries.

Using our efficient oracle-conversion technique between probability oracles and phase oracles, which incurs an $\widetilde{O}(1)$ overhead, the above lower bound implies an $\widetilde{\Omega}(\sqrt{d}/\varepsilon)$ query lower bound on ε-accurate gradient calculation for the probability oracle input model. □

Our lower bound for this family $\mathcal{F}$ shows that our gradient-calculation algorithm is in fact optimal up to poly-logarithmic factors for a specific class of smooth functions. We expect that our lower bound can be improved by extending its scope to a broader class of smooth functions with higher regularity, therefore showing optimality of our algorithm for a larger class of functions.

We are not aware of any prior work showing quantum query lower bounds on gradient-calculation. In fact most query lower bounds in quantum computing apply to settings where the input unitaries come from a discrete set which might correspond to some discrete computational problem. We know of only very few examples[9] where lower bounds are proven for a continuous set of unitary input oracles. The adversary method (discussed in Chapter 2), was also not long ago adapted to this continuous input setting by Belovs [Bel15].

5.4.3 Quantum variational eigensolvers and QAOA

In recent years, quantum adiabatic optimization algorithms (QAOA) [PMS+14, WHT15, FGG14] are favoured methods for providing low-depth quantum algorithms for solving important problems in quantum simulation and optimization. Current quantum computers are limited by decoherence, hence the option to solve optimization problems using very short circuits can be enticing even if such algorithms are polynomially more expensive than alternative strategies that could possibly require long gate sequences. Since these methods are typically envisioned as being appropriate only for low-depth applications, comparably less attention is paid to the question of what their complexity would be, if they were executed on a fault-tolerant quantum computer. In [GAW17], we consider the case that these algorithms are in fact implemented on a fault-tolerant quantum computer and show that the gradient calculation step in these algorithms can be performed quadratically faster compared to the earlier approaches that were tailored for pre-

[9] Quantum phase estimation is probably the best-known example.

fault-tolerant applications. Variational quantum eigensolvers (VQEs) are widely used to estimate the eigenvalue corresponding to some eigenstate of a Hamiltonian. The idea behind these approaches is to begin with an efficiently parameterizable ansatz to the eigenstate. For the example of ground state energy estimation, the ansatz state is often taken to be a unitary coupled cluster expansion. The terms in that unitary coupled cluster expansion are then varied to provide the lowest energy for the groundstate. Obtaining the optimal parameters in the ansatz involves a minimization problem. In [GAW17], we translate this optimization problem to one, which can be solved using our gradient-descent quantum algorithm and has quadratically better dependence on d.

5.4.4 Quantum auto-encoders

Classically, one application of neural networks is *auto-encoders*, which are networks that encode information about a data set into a low-dimensional representation. Auto-encoding was first introduced by Rumelhart et al. [RHW86]. Informally, the goal of an auto-encoding circuit is the following: given a set of high-dimensional vectors, the goal is to learn a representation of the vectors hopefully of low dimenension, so that computations on the original data set can be "approximately" carried out by working only with the low-dimensional vectors. More precisely the problem in auto-encoding is: Given $K < N$ and m data vectors $\{v_1, \ldots, v_m\} \subseteq \mathbb{R}^N$, find an encoding map $\mathcal{E} : \mathbb{R}^N \to \mathbb{R}^K$ and decoding map $\mathcal{D} : \mathbb{R}^K \to \mathbb{R}^N$ such that the average squared distortion $\|v_i - (\mathcal{D} \circ \mathcal{E})(v_i)\|^2$ is minimized:[10]

$$\min_{\mathcal{E},\mathcal{D}} \sum_{i \in [m]} \frac{\|v_i - (\mathcal{D} \circ \mathcal{E})(v_i)\|^2}{m}. \tag{5.17}$$

What makes auto-encoding interesting is that it does not assume any prior knowledge about the data set. This makes it a viable technique in machine learning, with various applications in natural language processing, training neural networks, object classification, prediction or extrapolation of information, etc.

Given that classical auto-encoders are 'work-horses' of classical machine learning [Azo94], it is also natural to consider a quantum variant of this paradigm. Very recently such quantum auto-encoding schemes have been proposed by Wan Kwak et al. [WDK+16] and independently by Romero et al. [ROA16]. Inspired by their work we provide a slightly generalized description of quantum auto-encoders by 'quantizing' auto-encoders the following way: we replace the data vectors v_i by quantum states ρ_i and define the maps $\mathcal{E}, \mathcal{D}$ as quantum channels transforming states back and forth between the Hilbert spaces. A natural generalization

[10] There are other natural choices of dissimilarity functions that one might want to minimize, for a comprehensive overview of the classical literature see [Bal12].

of squared distortion for quantum states ρ, σ that we consider is $1 - F^2(\rho, \sigma)$,[11] giving us the following minimization problem

$$\min_{\mathcal{E},\mathcal{D}} \sum_{i\in[m]} \frac{1 - F^2\left(\rho_i, (\mathcal{D}\circ\mathcal{E})(\rho_i)\right)}{m}. \tag{5.18}$$

When the input states are pure, i.e., $\rho_i = |\psi_i\rangle\langle\psi_i|$, then $F^2\left(|\psi\rangle\langle\psi|, \sigma\right) = \langle\psi|\sigma|\psi\rangle$. So, the above minimization problem is equivalent to the maximization problem

$$\max_{\mathcal{E},\mathcal{D}} \sum_{i\in[N]} \frac{\langle\psi_i|\left[(\mathcal{D}\circ\mathcal{E})(|\psi_i\rangle\langle\psi_i|)\right]|\psi_i\rangle}{m}. \tag{5.19}$$

Observe that $\langle\psi|\left[(\mathcal{D}\circ\mathcal{E})(|\psi\rangle\langle\psi|)\right]|\psi\rangle$ is the probability of finding the output state $(\mathcal{D}\circ\mathcal{E})(|\psi\rangle\langle\psi|)$ in state $|\psi\rangle$ after performing the projective measurement $\{|\psi\rangle\langle\psi|, I - |\psi\rangle\langle\psi|\}$. Thus we can think about this as maximizing the probability of recovering the initial pure state after encoding and decoding, which is a natural measure of the quality of the quantum auto-encoding procedure. In [GAW17], we translate this probability maximization problem to one, which can be solved using our gradient-descent quantum algorithm and has quadratically better dependence on d.

5.5 Conclusion and future work

We gave a new approach to quantum gradient calculation that is asymptotically optimal (up to logarithmic factors) for a class of smooth functions, in terms of the number of queries needed to estimate the gradient within fixed error with respect to the max-norm. This is based on several new ideas including the use of differentiation formula originating from high-degree interpolation polynomials. These high-degree methods quadratically improve the scaling of the query and time complexity with respect to the approximation quality compared to what one would see if the results from Jordan's work were used. We also provided lower bounds on the query complexity of the problem for certain smooth functions revealing that our algorithm is essentially optimal for a class of functions. While it has proven difficult to find natural applications for Jordan's original algorithm, we provide in this chapter several applications of our gradient-descent algorithm to areas ranging from machine learning to quantum chemistry simulation. These applications are built upon a method we provide for interconverting between phase and probability oracles. The polynomial speed-ups that we see for these

[11] Note that some authors (including [ROA16]) call $F'(\rho, \sigma) = F(\rho, \sigma)^2$ the *fidelity*, defined as $F(\rho, \sigma) = \mathrm{Tr}\left(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)$. The distortion measure we use here is $P(\rho, \sigma) = \sqrt{1 - F^2(\rho, \sigma)}$, which is called the purified (trace) distance [TCR10].

applications is made possible by our improved quantum gradient algorithm via the use of this interconversion process.

More work remains to be done in generalizing the lower bounds for functions that have stronger promises on the high-order derivatives. It would be interesting to see how quantum techniques can speed-up more sophisticated higher-level, e.g., stochastic gradient-descent methods. Another interesting question is whether quantum techniques can provide further speed-ups for calculating higher-order derivatives, such as the Hessian, using ideas related to Jordan's algorithm, see [Jor08, Appendix D]. Such improvements might open the door for improved quantum analogues of Newton's method and in turn substantially improve the scaling of the number of epochs needed to converge to a local optima in quantum methods. Another interesting direction is, whether we could improve the number of steps T in the gradient-descent algorithm? This improvement in T might be interesting and might have applications to *boosting* [FSA99] and quantum semi-definite programming (SDP) solvers [BS17, AGGW17].

Part Two

Learning in a quantum world

Chapter 6

Survey of quantum learning theory

This chapter is based on the paper "A survey of quantum learning theory", by S. Arunachalam and R. de Wolf [AW17a].

Abstract. In this chapter, we survey quantum learning theory: the theoretical aspects of machine learning using quantum computers. We describe the main results known for three models of learning: exact learning from membership queries, and Probably Approximately Correct (PAC) and agnostic learning from classical or quantum examples. In the first part of the chapter, we will consider the query complexity and sample complexity of learning algorithms and in the second part of the chapter we consider the time complexity of algorithms in these learning models.

Contents

6.1 Introduction

Machine learning entered theoretical computer science in the 1980s with the work of Leslie Valiant [Val84], who introduced the model of "Probably Approximately Correct" (PAC) learning, building on earlier work of Vapnik and others in statistics, but adding computational complexity aspects. This provided a mathematically rigorous definition of what it means to (efficiently) learn a target concept from given examples. In the three decades since, much work has been done in computational learning theory: some efficient learning results, many hardness results, and many more models of learning. We refer to [KV94b, AB09, SB14] for general introductions to this area. In recent years practical machine learning has gained an enormous boost from the success of deep learning in important big-data tasks like image recognition, natural language processing, and many other areas; this is theoretically still not very well understood, but it often works amazingly well.

Given the successes of both machine learning and quantum computing, combining these two strands of research is an obvious direction. Indeed, soon after Shor's algorithm, Bshouty and Jackson [BJ99] introduced a version of learning from *quantum* examples, which are quantum superpositions rather than random samples. They showed that Disjunctive Normal Form (DNF) can be learned efficiently from quantum examples under the uniform distribution; efficiently learning DNF from uniform *classical* examples (without membership queries) was and is an important open problem in classical learning theory. Servedio and others [AS05, AS09, SG04] studied upper and lower bounds on the number of quantum membership queries or quantum examples needed for learning, and more recently the author of this thesis and de Wolf obtained optimal bounds on quantum sample complexity [AW17b] (we discuss this in detail in the next chapter).

Much research in quantum machine learning has been on using quantum techniques to improve *specific* algorithms which are important in classical machine learning. In this direction, Aïmeur et al. [ABG06, ABG13] showed quantum speed-up in learning contexts such as clustering via minimum spanning

tree, divisive clustering, and k-medians, using variants of Grover's search algorithm [Gro96]. In the last few years there has been a flurry of interesting results applying various quantum algorithms (Grover's algorithm, but also phase estimation, amplitude amplification [BHMT02], and the HHL algorithm for solving well-behaved systems of linear equations [HHL09]) to machine learning problems. Examples include Principal Component Analysis [LMR13b], support vector machines [RML13], k-means clustering [LMR13a], quantum recommendation systems [KP17b], and more recent works related to neural networks [WKS16b, WKS16a]. Recently, there has also been work [GAW17] on using quantum techniques to improve gradient-based optimization algorithms which are ubiquitous in classical machine learning. Some of this work—like most of application-oriented machine learning in general—is heuristic in nature rather than mathematically rigorous. Some of these new approaches are suggestive of exponential speed-ups over classical machine learning, though one has to be careful about the underlying assumptions needed to make efficient quantum machine learning possible: in some cases these also make efficient *classical* machine learning possible. Aaronson [Aar15] gives a brief but clear description of the issues. These developments have been well-served by a number of recent survey papers [SSP15, AAD+15, BWP+17, CHI+17, DB17] and even a book [Wit14].

In contrast, in this chapter we focus on the theoretical side of quantum machine learning: quantum learning theory.[1] We will describe (and sketch proofs of) the main results that have been obtained in three main learning models. These will be described in much more detail in the next sections, but below we give a brief preview.

Exact learning. In this setting the goal is to learn a target concept from the ability to interact with it. For concreteness, we focus on learning target concepts that are Boolean functions $c : \{0,1\}^n \to \{0,1\}$. Considering concept classes over $\{0,1\}^n$ has the advantage that the n-bit x in a labeled example $(x, c(x))$ may be viewed as a "feature vector". This fits naturally when one is learning a type of objects characterized by patterns involving n features that each can be present or absent in an object, or when learning a class of n-bit Boolean functions such as small decision trees, circuits, or DNFs. However, we can (and sometimes do) also consider concepts $c : [N] \to \{0,1\}$.

In the framework of exact learning, the target concept is some unknown $c : \{0,1\}^n \to \{0,1\}$ coming from a known concept class $\mathcal{C}$ of functions, and our goal is to identify c exactly, with high probability, using *membership queries* (which allow the learner to learn $c(x)$ for x of his choice). If the measure of complexity is just the number of queries, the main results are that quantum exact

[1] The only other paper we are aware of to survey quantum learning theory is an unpublished manuscript by Robin Kothari from 2012 [Kot12] which is much shorter but partially overlaps with ours; we only saw this after finishing a first version of our survey [AW17a].

learners can be at most polynomially more efficient than classical, but not more. If the measure of complexity is *time*, then under reasonable complexity-theoretic assumptions some concept classes can be learned much faster from quantum membership queries (i.e., where the learner can query c on a superposition of x's) than is possible classically.

PAC learning. In this model of learning one also wants to learn an unknown $c : \{0,1\}^n \to \{0,1\}$ from a known concept class $\mathcal{C}$, but in a more passive way than with membership queries: the learner receives several *labeled examples* $(x, c(x))$, where x is distributed according to some unknown probability distribution D over $\{0,1\}^n$. The learner gets multiple i.i.d. labeled examples. From this limited "view" on c, the learner wants to generalize, producing a *hypothesis* h that probably agrees with c on "most" x, *measured according to the same* D. This is the classical Probably Approximately Correct (PAC) model. In the quantum PAC model [BJ99], an example is not a random sample but a *superposition* $\sum_{x\in\{0,1\}^n} \sqrt{D(x)}|x, c(x)\rangle$. Such quantum examples can be useful for some learning tasks with a fixed distribution D (e.g., uniform D) but it turns out that in the usual distribution-independent PAC model, quantum and classical sample complexity are equal up to constant factors, for every concept class $\mathcal{C}$. When the measure of complexity is *time*, under reasonable complexity-theoretic assumptions, some concept classes can be PAC learned much faster by quantum learners (even from classical examples) than is possible classically.

Agnostic learning. In this setting, a learner wants to approximate a distribution on $\{0,1\}^{n+1}$ by finding a good hypothesis h to predict the last bit from the first n bits. A "good" hypothesis is one that is not much worse than the best predictor available in a given class $\mathcal{C}$ of available hypotheses. The agnostic model has more freedom than the PAC model and allows to model more realistic situations, for example when the data is noisy or when no "perfect" target concept exists. Like in the PAC model, it turns out quantum sample complexity is not significantly smaller than classical sample complexity in the agnostic model.

Organization. This chapter is organized as follows. In Section 6.2 we first introduce a few quantum subroutines that we use often. In Section 6.3, we introduce the classical and quantum learning models. In Section 6.4 and 6.5, we describe the main results obtained for information-theoretic measures of learning complexity, namely query complexity of exact learning and sample complexities of PAC and agnostic learning. In Section 6.6 we discuss known results in learning quantum objects. In Section 6.7 we survey the main results known about *time* complexity of quantum learners. We conclude in Section 6.8 with a summary of the results and some open questions for further research.

6.2 Quantum subroutines

6.2.1 Grover's algorithm

We have already discussed Grover's search algorithm in Chapter 2, for finding a solution in N-bit database using $O(\sqrt{N})$ quantum queries in Section 2.5. In this chapter, we will invoke the following more recent application of Grover's algorithm.

6.2.1. THEOREM ([Kot14],[LL16, Theorem 5.6]). *Suppose $x \in \{0,1\}^N$. There exists a quantum algorithm that satisfies the following properties:*

- *if $x \neq 0^N$, then let d be the first (i.e., smallest) index satisfying $x_d = 1$; the algorithm uses an expected number of $O(\sqrt{d})$ queries to x and outputs d with probability at least $2/3$;*
- *if $x = 0^N$ then the algorithm always outputs "no solution" after $O(\sqrt{N})$ queries.*

6.2.2 Fourier sampling

A very simple but powerful quantum algorithm is *Fourier sampling*. An introduction to Fourier analysis can be found in Section 2.1. Consider a function $f : \{0,1\}^n \to \mathbb{R}$. The Fourier decomposition of f is $f = \sum_S \widehat{f}(S)\chi_S$. Parseval's identity says that $\sum_S \widehat{f}(S)^2 = \mathbb{E}_x[f(x)^2]$. Note that if f has range $\{\pm 1\}$ then Parseval implies that the squared Fourier coefficients $\widehat{f}(S)^2$ sum to 1, and hence form a probability distribution. Fourier sampling means sampling an S with probability $\widehat{f}(S)^2$. Classically this is a hard problem, because the probabilities $\widehat{f}(S)^2$ depend on all 2^n values of f (since $\widehat{f}(S) = \frac{1}{2^n}\sum_x f(x)\chi_S(x)$). However, the following quantum algorithm due to Bernstein and Vazirani [BV97] does it exactly using *only* 1 query and $O(n)$ gates.

1. Start with $|0^n\rangle$.
2. Apply Hadamard transforms to all n qubits, obtaining $\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle$.
3. Query O_f,[2] obtaining $\frac{1}{\sqrt{2^n}}\sum_x f(x)|x\rangle$.
4. Apply Hadamard transforms to all n qubits to obtain
$$\frac{1}{\sqrt{2^n}}\sum_x f(x)\Big(\frac{1}{\sqrt{2^n}}\sum_S(-1)^{x\cdot S}|S\rangle\Big) = \sum_S \widehat{f}(S)|S\rangle.$$
5. Measure the state, obtaining S with probability $\widehat{f}(S)^2$.

[2]Here we view $f \in \{1,-1\}^{2^n}$ as being specified by its truth-table, so $O_f : |x\rangle \to f(x)|x\rangle$.

6.3 Learning models

In this section we will define the three main learning models that we focus on: the *exact* model of learning introduced by Angluin [Ang87], the *PAC* model of learning introduced by Valiant [Val84], and the *agnostic* model of learning introduced by Haussler [Hau92] and Kearns et al. [KSS94]. Below, a *concept class* $\mathcal{C}$ will usually be a set of functions $c : \{0,1\}^n \to \{0,1\}$, though we can also allow functions $c : [N] \to \{0,1\}$, or treat such a c as an N-bit string specified by its truth-table.

6.3.1 Exact learning

Classical exact learning. In the exact learning model, a learner $\mathcal{A}$ for a concept class $\mathcal{C}$ is given access to a *membership oracle* $\mathrm{MQ}(c)$ for the unknown *target concept* $c \in \mathcal{C}$ that $\mathcal{A}$ is trying to learn. Given an input $x \in \{0,1\}^n$, $\mathrm{MQ}(c)$ returns the label $c(x)$. A learning algorithm $\mathcal{A}$ is an *exact learner* for $\mathcal{C}$ if:

> For every $c \in \mathcal{C}$, given access to the $\mathrm{MQ}(c)$ oracle:
> with probability at least $2/3$, $\mathcal{A}$ outputs an h such that $h(x) = c(x)$ for all $x \in \{0,1\}^n$.[3]

This model is also sometimes known as "oracle identification": the idea is that $\mathcal{C}$ is a set of possible oracles, and we want to efficiently identify which $c \in \mathcal{C}$ is our actual oracle, using membership queries to c. The *query complexity* of $\mathcal{A}$ is the maximum number of invocations of the $\mathrm{MQ}(c)$ oracle which the learner makes, over all concepts $c \in \mathcal{C}$ and over the internal randomness of the learner. The *query complexity of exactly learning* $\mathcal{C}$ is the minimum query complexity over all exact learners for $\mathcal{C}$.[4]

Each concept $c : \{0,1\}^n \to \{0,1\}$ can also be specified by its N-bit truth-table (with $N = 2^n$), hence one may view the concept class $\mathcal{C}$ as a subset of $\{0,1\}^N$. For a given N and M, define the (N, M)-*query complexity of exact learning* as the maximum query complexity of exactly learning $\mathcal{C}$, maximized over all $\mathcal{C} \subseteq \{0,1\}^N$ such that $|\mathcal{C}| = M$.

Quantum exact learning. In the quantum setting, instead of having access to an $\mathrm{MQ}(c)$ oracle, a *quantum exact learner* is given access to a $\mathrm{QMQ}(c)$ oracle, which corresponds to the map $\mathrm{QMQ}(c) : |x, b\rangle \to |x, b \oplus c(x)\rangle$ for $x \in \{0,1\}^n, b \in \{0,1\}$. For a given $\mathcal{C}, N, M$, one can define the *quantum query complexity of*

[3] We could also consider a δ-exact learner who succeeds with probability $1 - \delta$, but here restrict to $\delta = 1/3$ for simplicity. Standard amplification techniques can reduce this $1/3$ to any $\delta > 0$ at the expense of an $O(\log(1/\delta))$ factor in the complexity.

[4] This terminology of "learning $\mathcal{C}$" or "$\mathcal{C}$ is learnable" is fairly settled though slightly unfortunate: what is actually being learned is of course a target concept $c \in \mathcal{C}$, not the class $\mathcal{C}$ itself, which the learner already knows from the start.

exactly learning $\mathcal{C}$, and the (N, M)*-quantum query complexity of exact learning* as the quantum analogues (where the learner is given access to the QMQ(c) oracle) to the classical complexity measures.

6.3.2 Probably approximately correct (PAC) learning

Classical PAC model. In this section we will be concerned mainly with the PAC (Probably Approximately Correct) model of learning introduced by Valiant [Val84]. For further reading, see standard textbooks in computational learning theory such as [KV94b, AB09, SB14]. In the classical PAC model, a learner $\mathcal{A}$ is given access to a *random example oracle* PEX(c, D) which generates labeled examples of the form $(x, c(x))$ where x is drawn from an unknown distribution $D : \{0,1\}^n \to [0,1]$ and $c \in \mathcal{C}$ is the *target concept* that $\mathcal{A}$ is trying to learn. For a concept $c \in \mathcal{C}$ and hypothesis $h : \{0,1\}^n \to \{0,1\}$, we define the error of h compared to the target concept c, under D, as

$$\mathrm{err}_D(h, c) = \Pr_{x \sim D}[h(x) \neq c(x)].$$

A learning algorithm $\mathcal{A}$ is an (ε, δ)*-PAC learner* for $\mathcal{C}$, if the following holds:

> For every $c \in \mathcal{C}$, distribution D, given access to the PEX(c, D) oracle:
> $\mathcal{A}$ outputs an h such that $\mathrm{err}_D(h, c) \leq \varepsilon$ with probability at least $1 - \delta$.

Note that the learner has the freedom to output an hypothesis h which is not itself in the concept class $\mathcal{C}$. If the learner always produces an $h \in \mathcal{C}$, then it is called a *proper* PAC learner.

The *sample complexity* of $\mathcal{A}$ is the maximum number of invocations of the PEX(c, D) oracle which the learner makes, over all concepts $c \in \mathcal{C}$, distributions D, and the internal randomness of the learner. Clearly there are different PAC-learners that learn $\mathcal{C}$. The (ε, δ)*-PAC sample complexity* of a concept class $\mathcal{C}$ is the minimum sample complexity over all (ε, δ)-PAC learners for $\mathcal{C}$.

Quantum PAC model. The quantum PAC learning model was introduced by Bshouty and Jackson in [BJ99]. The quantum PAC model is a generalization of the classical PAC model, instead of having access to random examples $(x, c(x))$ from the PEX(c, D) oracle, the learner now has access to superpositions over all $(x, c(x))$. For an unknown distribution $D : \{0,1\}^n \to [0,1]$ and concept $c \in \mathcal{C}$, a *quantum example oracle* QPEX(c, D) acts on $|0^n, 0\rangle$ and produces a *quantum example*

$$\sum_{x \in \{0,1\}^n} \sqrt{D(x)} |x, c(x)\rangle,$$

we leave QPEX undefined on other basis states. Such a quantum example is the natural quantum generalization of a classical random sample.[5] While it is

[5] We could also allow complex phases for the amplitudes $\sqrt{D(x)}$; however, these will make no difference for the results presented here.

not always realistic to assume access to such (fragile) quantum states, one can certainly envision learning situations where the data is provided by a coherent quantum process.

A quantum learner is given access to some copies of the state generated by $\mathrm{QPEX}(c, D)$ and performs a POVM where each outcome is associated with a hypothesis. A learning algorithm $\mathcal{A}$ is an (ε, δ)-*PAC quantum learner* for $\mathcal{C}$ if:

> For every $c \in \mathcal{C}$, distribution D, given access to the $\mathrm{QPEX}(c, D)$ oracle:
> $\mathcal{A}$ outputs an h such that $\mathrm{err}_D(h, c) \leq \varepsilon$, with probability at least $1-\delta$.

The *sample complexity* of the learning algorithm $\mathcal{A}$ is the maximum number invocations of the $\mathrm{QPEX}(c, D)$ oracle, maximized over all $c \in \mathcal{C}$, distributions D, and the learner's internal randomness. The (ε, δ)*-PAC quantum sample complexity* of a concept class $\mathcal{C}$ is the minimum sample complexity over all (ε, δ)-PAC quantum learners for $\mathcal{C}$.

Observe that from a quantum example $\sum_x \sqrt{D(x)}|x, c(x)\rangle$, we can obtain $\sum_x \sqrt{D(x)}(-1)^{c(x)}|x\rangle$ with probability $1/2$: apply the Hadamard transform to the last qubit and measure it. With probability $1/2$ we obtain the outcome 1, in which case the remaining state is $\sum_x \sqrt{D(x)}(-1)^{c(x)}|x\rangle$. If D is the uniform distribution, then the obtained state is exactly the state needed in step 3 of the Fourier sampling algorithm described in Section 6.2.2.

How does the model of quantum examples compare to the model of quantum membership queries? If the distribution D is known, a membership query can be used to create a quantum example: the learner can create the superposition $\sum_x \sqrt{D(x)}|x, 0\rangle$ and apply a membership query to the target concept c to obtain a quantum example. On the other hand, as Bshouty and Jackson [BJ99] already observed, a membership query *cannot* be simulated using a small number of quantum examples. Consider for example the learning problem corresponding to Grover search, where the concept class $\mathcal{C} \subseteq \{0,1\}^N$ consists of all strings of weight 1, i.e., $\mathcal{C} = \{e_i : i \in [N]\}$. We know that $\Theta(\sqrt{N})$ quantum membership queries are necessary and sufficient to exactly learn the target concept with high probability. However, it is not hard to show that, under the uniform distribution, one needs $\Omega(N)$ quantum examples to exactly learn the target concept with high probability. Hence simulating one membership query requires at least $\Omega(\sqrt{N})$ quantum examples.

6.3.3 Agnostic learning

Classical agnostic model. The PAC model assumes that the labeled examples are generated perfectly according to a target concept $c \in \mathcal{C}$. However, in many learning situations that is not a realistic assumption, for example when the examples are noisy in some way or when we have no reason to believe there is an underlying target concept at all. The agnostic model of learning introduced by Haussler [Hau92] and Kearns et al. [KSS94], takes this into account.

Agnostic learning is the following: for a distribution $D : \{0,1\}^{n+1} \to [0,1]$, a learner $\mathcal{A}$ is given access to an $\mathrm{AEX}(D)$ oracle that generates examples of the form (x, b) drawn from the distribution D. We define the error of $h : \{0,1\}^n \to \{0,1\}$ under D as

$$\mathrm{err}_D(h) = \Pr_{(x,b)\sim D}[h(x) \neq b].$$

When h is restricted to come from a concept class $\mathcal{C}$, the minimal error achievable is $\mathrm{opt}_D(\mathcal{C}) = \min_{c\in\mathcal{C}}\{\mathrm{err}_D(c)\}$.

In agnostic learning, a learner $\mathcal{A}$ needs to output a hypothesis h whose error is not much bigger than $\mathrm{opt}_D(\mathcal{C})$. A learning algorithm $\mathcal{A}$ is an (ε, δ)-*agnostic learner* for $\mathcal{C}$ if:

> For every distribution D on $\{0,1\}^{n+1}$, given access to $\mathrm{AEX}(D)$:
> $\mathcal{A}$ outputs an $h \in \mathcal{C}$ such that $\mathrm{err}_D(h) \leq \mathrm{opt}_D(\mathcal{C}) + \varepsilon$ with probability at least $1 - \delta$.

Note that if there is a $c \in \mathcal{C}$ which perfectly classifies every x with label y for $(x, y) \in \mathrm{supp}(D)$, then $\mathrm{opt}_D(\mathcal{C}) = 0$ and we are in the setting of proper PAC learning. The *sample complexity* of $\mathcal{A}$ is the maximum number of invocations of the $\mathrm{AEX}(c, D)$ oracle which the learner makes, over all distributions D and over the learner's internal randomness. The (ε, δ)-*agnostic sample complexity* of a concept class $\mathcal{C}$ is the minimum sample complexity over all (ε, δ)-agnostic learners for $\mathcal{C}$.

Quantum agnostic model. The model of quantum agnostic learning was first studied in [AW17b]. For a joint distribution $D : \{0,1\}^{n+1} \to [0,1]$ over the set of examples, the learner has access to an $\mathrm{QAEX}(D)$ oracle which acts on $|0^n, 0\rangle$ and produces a quantum example $\sum_{(x,b)\in\{0,1\}^{n+1}} \sqrt{D(x,b)}|x,b\rangle$ (we again leave $\mathrm{QAEX}(D)$ undefined on other basis states). A learning algorithm $\mathcal{A}$ is an (ε, δ)-*agnostic quantum learner* for $\mathcal{C}$ if:

> For every distribution D, given access to the $\mathrm{QAEX}(D)$ oracle:
> $\mathcal{A}$ outputs an $h \in \mathcal{C}$ such that $\mathrm{err}_D(h) \leq \mathrm{opt}_D(\mathcal{C}) + \varepsilon$ with probability at least $1 - \delta$.

The *sample complexity* of $\mathcal{A}$ is the maximum number invocations of the $\mathrm{QAEX}(D)$ oracle over all distributions D and over the learner's internal randomness. The (ε, δ)-*agnostic quantum sample complexity* of a concept class $\mathcal{C}$ is the minimum sample complexity over all (ε, δ)-agnostic quantum learners for $\mathcal{C}$.

6.4 Results on query complexity

In this section, we begin by proving bounds on the quantum query complexity of exactly learning a concept class $\mathcal{C}$ in terms of a combinatorial parameter $\gamma(\mathcal{C})$,

which we define shortly, and then sketch the proof of optimal bounds on (N, M)-quantum query complexity of exact learning.

Throughout this section, we will specify a concept $c : \{0,1\}^n \to \{0,1\}$ by its N-bit truth-table (with $N = 2^n$), hence $\mathcal{C} \subseteq \{0,1\}^N$. For a set $S \subseteq \{0,1\}^N$, we will use the "N-bit majority string" $\mathrm{MAJ}(S) \in \{0,1\}^N$ defined as: $\mathrm{MAJ}(S)_i = 1$ iff $|\{s \in S : s_i = 1\}| \geq |\{s \in S : s_i = 0\}|$.

6.4.1. Definition. (Combinatorial parameter $\gamma(\mathcal{C})$) Let $\mathcal{C} \subseteq \{0,1\}^N$ be a concept class of size $|\mathcal{C}| > 1$, and let $\mathcal{C}' \subseteq \mathcal{C}$. For $i \in [N]$ and $b \in \{0,1\}$, define

$$\gamma'(\mathcal{C}', i, b) = \frac{|\{c \in \mathcal{C}' : c_i = b\}|}{|\mathcal{C}'|}$$

as the fraction of concepts in $\mathcal{C}'$ that satisfy $c_i = b$. Let

$$\gamma'(\mathcal{C}', i) = \min\{\gamma'(\mathcal{C}', i, 0), \gamma'(\mathcal{C}', i, 1)\}$$

be the minimum fraction of concepts that can be eliminated by learning c_i. Let

$$\gamma'(\mathcal{C}') = \max_{i \in [N]}\{\gamma'(\mathcal{C}', i)\}$$

denote the largest fraction of concepts in $\mathcal{C}'$ that can be eliminated by a query. Finally, define

$$\gamma(\mathcal{C}) = \min_{\substack{\mathcal{C}' \subseteq \mathcal{C}, \\ |\mathcal{C}'| \geq 2}} \gamma'(\mathcal{C}') = \min_{\substack{\mathcal{C}' \subseteq \mathcal{C}, \\ |\mathcal{C}'| \geq 2}} \max_{i \in [N]} \min_{b \in \{0,1\}} \gamma'(\mathcal{C}', i, b).$$

This complicated-looking definition is motivated by the following learning algorithm. Suppose the learner wants to exactly learn an unknown $c \in \mathcal{C}$. Greedily, the learner would query c on the "best" input $i \in [N]$, i.e., the i that eliminates the largest fraction of concepts from $\mathcal{C}$ irrespective of the value of c_i. Suppose j is the "best" input (i.e., $i = j$ maximizes $\gamma'(\mathcal{C}, i)$) and the learner queries c on index j, then at least a $\gamma(\mathcal{C})$-fraction of the concepts in $\mathcal{C}$ will be inconsistent with the query-outcome, and these can now be eliminated from $\mathcal{C}$. Call the set of remaining concepts $\mathcal{C}'$, and note that $|\mathcal{C}'| \leq (1 - \gamma(\mathcal{C}))|\mathcal{C}|$. The outermost min in $\gamma(\mathcal{C})$ guarantees that there will be another query that the learner can make to eliminate at least a $\gamma(\mathcal{C})$-fraction of the remaining concepts from $\mathcal{C}'$, and so on. We stop when there is only one remaining concept left. Since each query will shrink the set of remaining concepts by a factor of at least $1 - \gamma(\mathcal{C})$, making $T = O((\log |\mathcal{C}|)/\gamma(\mathcal{C}))$ queries suffices to shrink $\mathcal{C}$ to $\{c\}$.

6.4.1 Complexity of exactly learning $\mathcal{C}$ in terms of $\gamma(\mathcal{C})$

Bshouty et al. [BCG+96] showed the following bounds on the classical complexity of exactly learning a concept class $\mathcal{C}$ (we already sketched the upper bound above).

6.4.2. THEOREM ([BCG+96, SG04]). *Every classical exact learner for concept class $\mathcal{C}$ has to use $\Omega(\max\{1/\gamma(\mathcal{C}), \log|\mathcal{C}|\})$ membership queries. For every $\mathcal{C}$, there is a classical exact learner which learns $\mathcal{C}$ using $O(\frac{\log|\mathcal{C}|}{\gamma(\mathcal{C})})$ membership queries.*

In order to show a polynomial relation between quantum and classical exact learning, Servedio and Gortler [SG04] showed the following lower bounds.

6.4.3. THEOREM ([SG04]). *Let $N = 2^n$. Every quantum exact learner for concept class $\mathcal{C} \subseteq \{0,1\}^N$ has to make $\Omega(\max\{\frac{1}{\sqrt{\gamma(\mathcal{C})}}, \frac{\log|\mathcal{C}|}{n}\})$ membership queries.*

Proof sketch. We first prove the $\Omega(1/\sqrt{\gamma(\mathcal{C})})$ lower bound. We will use the positive-weight adversary bound of Ambainis [Amb00], discussed in Section 2.4.2. In order to put this bound to use, we need to construct a relation $R \subseteq \mathcal{D} \times \mathcal{D}$, that maximizes $\Omega(\sqrt{mm'/\ell\ell'})$, where $\mathcal{D} \subseteq \{0,1\}^N$.

Now we want to apply this lower bound to a quantum exact learner for concept class $\mathcal{C}$. We can think of the learning algorithm as making queries to an N-bit input string and producing the name of a concept $c \in \mathcal{C}$ as output. Suppose $\mathcal{C}' \subseteq \mathcal{C}$ is a minimizer in the definition of $\gamma(\mathcal{C})$ (i.e., $\gamma'(\mathcal{C}') = \gamma(\mathcal{C})$). Define $\tilde{c} = \mathrm{MAJ}(\mathcal{C}')$. Note that $\tilde{c}$ need not be in $\mathcal{C}'$ or even in $\mathcal{C}$, but we can still consider what our learner does on input $\tilde{c}$. We consider two cases:

Case 1: For every $c \in \mathcal{C}'$, the probability that the learner outputs c when run on the typical concept $\tilde{c}$, is $< 1/2$. In this case we pick our relation $R = \{\tilde{c}\} \times \mathcal{C}'$. Calculating the parameters for the adversary bound, we have $m = |\mathcal{C}'|$, $m' = 1$, $\ell \leq \gamma'(\mathcal{C}')|\mathcal{C}'|$ (because for every i, $\tilde{c}_i \neq c_i$ for a $\gamma'(\mathcal{C}', i)$-fraction of the $c \in \mathcal{C}'$ and $\gamma'(\mathcal{C}', i) \leq \gamma'(\mathcal{C}')$ by definition), and $\ell' = 1$. Since, for every $c \in \mathcal{C}'$, the learner outputs c with high probability on input c, the final states on every pair of R-related concepts will be $\Omega(1)$ apart. Using Theorem 2.4.5, the number of queries that our learner needs to make is $\Omega(\sqrt{mm'/\ell\ell'}) = \Omega(1/\sqrt{\gamma'(\mathcal{C}')}) = \Omega(1/\sqrt{\gamma(\mathcal{C})})$ (because $\mathcal{C}'$ minimized $\gamma(\mathcal{C})$).

Case 2: When $\tilde{c}$, there exists a specific $c \in \mathcal{C}'$ that the learner gives as output with probability $\geq 1/2$ when run on input $\tilde{c}$. In this case we pick $R = \{\tilde{c}\} \times (\mathcal{C}' \backslash \{c\})$, ensuring that the final states on every pair of R-related concepts will be $\Omega(1)$ apart. We now have $m = |\mathcal{C}'| - 1$, $m' = 1$, $\ell \leq \gamma'(\mathcal{C}')|\mathcal{C}'|$ (for the same reason as in Case 1), and $\ell' = 1$. Since $(|\mathcal{C}'| - 1)/|\mathcal{C}'| = \Omega(1)$, the adversary bound again yields an $\Omega(1/\sqrt{\gamma(\mathcal{C})})$ bound.

We now prove the $\Omega((\log|\mathcal{C}|)/n)$ lower bound by an information-theoretic argument, as follows. View the target string $c \in \mathcal{C}$ as a uniformly distributed random variable. If our algorithm can exactly identify c with high success probability, it has learned $\Omega(\log|\mathcal{C}|)$ bits of information about c (formally, the mutual information between c and the learner's output is $\Omega(\log|\mathcal{C}|)$). From Holevo's theorem [Hol73], since a quantum query acts on only $n+1$ qubits, one quantum query can yield at most $O(n)$ bits of information about c. Hence $\Omega((\log|\mathcal{C}|)/n)$

quantum queries are needed. □

Both of the above lower bounds are in fact individually optimal. First, if one takes $\mathcal{C} \subseteq \{0,1\}^N$ to consist of the N functions c for which $c(i) = 1$ for exactly one i, then exact learning corresponds to the unordered search problem with 1 solution. Here $\gamma(\mathcal{C}) = 1/N$, and $\Theta(\sqrt{N})$ queries are necessary and sufficient thanks to Grover's algorithm. Second, if $\mathcal{C}$ is the class of $N = 2^n$ linear functions on $\{0,1\}^n$, $\mathcal{C} = \{c(x) = a \cdot x : a \in \{0,1\}^n\}$, then Fourier sampling gives an $O(1)$-query algorithm (see Section 6.7.3). In addition to these quantum-classical separations based on Grover and Fourier sampling, in Section 6.7.3 we also mention a fourth-power separation between $Q(\mathcal{C})$ and $D(\mathcal{C})$ due to Belovs [Bel13], for the problem of learning certain *k-juntas*. Combining Theorems 6.4.2 and 6.4.3, Servedio and Gortler [SG04] showed that the classical and quantum query complexity of exact learning are essentially polynomially related for every $\mathcal{C}$.

6.4.4. Corollary ([SG04]). *If concept class $\mathcal{C}$ has classical and quantum membership query complexities $D(\mathcal{C})$ and $Q(\mathcal{C})$, respectively, then $D(\mathcal{C}) = O(nQ(\mathcal{C})^3)$.*

6.4.5. Remark. In work in progress [ACLW18], we improved this upper bound by a logarithmic factor. In particular, we showed that for a concept class $\mathcal{C} \subseteq \{0,1\}^N$, we have

$$D(\mathcal{C}) \leq O\Big(\frac{Q(\mathcal{C})^2 \log |\mathcal{C}|}{\log Q(\mathcal{C}) + 1}\Big). \tag{6.1}$$

Observe that this bound is tight for the concept class $\mathcal{C} = \{e_i : i \in [N]\}$, which satisfies $D(\mathcal{C}) = N$, $Q(\mathcal{C}) = \Theta(\sqrt{N})$ and $|\mathcal{C}| = N$ and for the concept class of linear functions $\mathcal{C} = \{c(x) = a \cdot x : a \in \{0,1\}^n\}$, which satisfies $D(\mathcal{C}) = n$, $Q(\mathcal{C}) = 1$ and $|\mathcal{C}| = 2^n$. Combining with Theorem 6.4.3, our upper bound in Eq. (6.1) yields $D(\mathcal{C}) \leq O(nQ(\mathcal{C})^3/\log Q(\mathcal{C}))$, improving upon the upper bound of Servedio and Gortler [SG04, Theorem 1.1].

We briefly give the proof idea here. We first construct a distributional learner such that: for every distribution μ on $\mathcal{C}$, the μ-distributional learner has success probability at least $2/3$ measured under μ. The queries made by the learner is chosen as follows. We use the negative-weight adversary method to show the existence of an index $i \in [N]$ that satisfies the following: suppose the learner queries the unknown concept at index i, then the number of concepts in $\mathcal{C}$ consistent with the query outcome reduces at least by a $(1 - \frac{1}{Q(\mathcal{C})^2})$-factor. We then define $\mathcal{C}'$ by restricting to the concepts in $\mathcal{C}$ consistent with the query outcome. We again use the adversary method to find an index i' which the learner should query and define $\mathcal{C}''$ based on the query outcome at index i'. We repeat this process until a unique concept remains. Using an information-theoretic argument, we show that repeating this process $O(\frac{Q(\mathcal{C})^2 \log |\mathcal{C}|}{\log Q(\mathcal{C})+1})$ times suffice to find the unknown

concept. Eq. (6.1) follows for distributional learners because each round uses at most one membership query. In order to conclude the proof, we invoke the Yao principle [Yao77] to show the existence of a classical learner who, on input $c \in \mathcal{C}$, outputs the label for c with probability at least $2/3$.

6.4.2 (N, M)-query complexity of exact learning

In this section we focus on the (N, M)-quantum query complexity of exact learning. Classically, the following characterization is easy to prove.

6.4.6. Theorem (Folklore). *The (N, M)-query complexity of exact learning is* $\Theta(\min\{M, N\})$.

Proof sketch. Clearly N is an upper bound since $\mathcal{C} \subseteq \{0,1\}^N$. For $M \leq N$, using one query we can eliminate at least one concept from $\mathcal{C}$, so M queries suffice. For the lower bounds, consider the concept class $\mathcal{C}' = \{e_i : i \in [N]\}$. Clearly we need $\Omega(N)$ queries to exactly learn a $c \in \mathcal{C}$. Suppose $M \leq N$, just pick a subset of $\mathcal{C}'$ of size M and the query complexity of exactly learning this subset of $\mathcal{C}'$ is at least $\Omega(M)$. □

In the quantum context, the (N, M)-query complexity of exact learning has been completely characterized by Kothari [Kot14]. He improved upon a sequence of works [AIK+04, AIK+07, AIN+09], and showed the following theorem.

6.4.7. Theorem ([Kot14]). *The (N, M)-quantum query complexity of exact learning is $\Theta(\sqrt{M})$ for $M \leq N$ and $\Theta\left(\sqrt{\frac{N \log M}{\log(N/\log M)+1}}\right)$ for $N < M \leq 2^N$.*

Proof sketch. We first sketch the lower bound proofs before moving on to showing the upper bound.

Lower bounds. Consider first the case $M \leq N$. Suppose $\mathcal{C} \subseteq \{c \in \{0,1\}^N : |c| = 1\}$ satisfies $|\mathcal{C}| = M$. Then, exactly learning $\mathcal{C}$ is as hard as the unordered search problem on M bits, which requires $\Omega(\sqrt{M})$ quantum queries. For the case $N < M \leq 2^N$, we need the following lemma.

6.4.8. Lemma. *There exists $\mathcal{C} \subseteq \{0,1\}^N$ of size $|\mathcal{C}| \leq M$, such that the query complexity of exactly learning $\mathcal{C}$ is $\Omega(\sqrt{(N-k+1)k})$ for every k that satisfies $\binom{N}{k-1} + \binom{N}{k} \leq M$.*

Proof. Let $\mathcal{C} \subseteq \{0,1\}^N$ be the set of N-bit strings with Hamming weight $k-1$ or k (for some k satisfying $\binom{N}{k-1} + \binom{N}{k} \leq M$, so that $|\mathcal{C}| \leq M$). Suppose $\mathcal{A}$ is an quantum exact learning algorithm for $\mathcal{C}$. Being able to learn a concept $c \in \mathcal{C}$, in particular implies that $\mathcal{A}$ can distinguish between two concepts x and y such that

$|x| = k$ and $|y| = k-1$. Using the positive-weight adversary method, it follows that every $\mathcal{A}$ satisfying this property has to make $\Omega(\sqrt{(N-k+1)k})$ queries to the unknown concept.[6] □

The fact below shows that a sufficiently large k exists satisfying the requirement of the lemma above.

6.4.9. Fact. For every $N < M \leq 2^N$, there exists $k \in \Omega\left(\frac{\log M}{\log(N/\log M)+1}\right)$ such that $\binom{N}{k-1} + \binom{N}{k} \leq M$.

The proof of this combinatorial fact is not too hard and we refer the reader to [Kot14, Lemma 5]. Combining this fact along with Lemma 6.4.8, the lower bound in the theorem follows.

Upper bounds. We now sketch the proofs of the upper bound. We use the following notation: for $u \in \{0,1\}^n$ and $S \subseteq [n]$, let $u_S \in \{0,1\}^{|S|}$ be the string obtained by restricting u to the indices in S.

We first describe a quantum algorithm that gives a worse upper bound than promised, but is easy to explain. Suppose $\mathcal{C} \subseteq \{0,1\}^N$ satisfies $|\mathcal{C}| = M$. Let $c \in \mathcal{C}$ be the unknown target concept that the algorithm is trying to learn. The basic idea of the algorithm is as follows: use the algorithm of Theorem 6.2.1 to find the first index $p_1 \in [N]$ at which c and $\mathrm{MAJ}(\mathcal{C})$ differ. This uses an expected $O(\sqrt{p_1})$ quantum queries to c (if there is no difference, i.e., $c = \mathrm{MAJ}(\mathcal{C})$, then the algorithm will tell us so after $O(\sqrt{N})$ queries and we can stop). We have now learned the first p_1 bits of c. Let $\mathcal{C}_1 = \{z_{[N]\setminus[p_1]} : z \in \mathcal{C},\ z_{[p_1-1]} = \mathrm{MAJ}(\mathcal{C})_{[p_1-1]},\ z_{p_1} = \overline{\mathrm{MAJ}(\mathcal{C})}_{p_1}\} \subseteq \{0,1\}^{N-p_1}$ be the set of suffixes of the concepts in $\mathcal{C}$ that agree with $\mathrm{MAJ}(\mathcal{C})$ on the first $p_1 - 1$ indices and disagree with $\mathrm{MAJ}(\mathcal{C})$ on the p_1th index. Similarly, let $c^1 = c_{[N]\setminus[p_1]}$ be the "updated" unknown target concept after restricting c to the coordinates $\{p_1+1,\ldots,N\}$. Next, we use the same idea to find the first index $p_2 \in [N-p_1]$ such that $(c^1)_{p_2} \neq \mathrm{MAJ}(\mathcal{C}_1)_{p_2}$. Repeat this until only one concept is left, and let r be the number of repetitions (i.e., until $|\mathcal{C}_r| = 1$).

In order to analyze the query complexity, first note that, for $k \geq 1$, the k-th iteration of the procedure gives us p_k bits of c. Since the procedure repeated r times, we have $p_1 + \cdots + p_r \leq N$. Second, each repetition in the algorithm reduces the size of $\mathcal{C}_i$ by at least a half, i.e., for $i \geq 2$, $|\mathcal{C}_i| \leq |\mathcal{C}_{i-1}|/2$. Hence one needs to repeat the procedure at most $r \leq O(\log M)$ times. The last run will use

[6] In order to apply the positive-weight adversary method (Theorem 2.4.5), we could use the relation $R \subseteq f^{-1}(0) \times f^{-1}(1)$ defined as: $R = \{(x, x+e_i) : x \in \{0,1\}^N, |x| = k-1, i \notin \mathrm{supp}(x)\}$. For this R, observe that $m = N-k+1, m' = k, \ell = \ell' = 1$, which gives us the $\Omega(\sqrt{(N-k+1)k})$ lower bound.

$O(\sqrt{N})$ queries and will tell us that we have learned all the bits of c. It follows that the total number of queries the algorithm makes to c is

$$\sum_{k=1}^{r} O(\sqrt{p_k}) + O(\sqrt{N}) \leq O\left(\sqrt{r}\sqrt{\sum_{k=1}^{r} p_k}\right) + O(\sqrt{N}) \leq O(\sqrt{N \log M}),$$

where we used the Cauchy-Schwarz inequality first and our upper bounds on $r \leq O(\log M)$ and $\sum_i p_i \leq N$ in the second inequality.[7]

This algorithm is an $O(\sqrt{\log(N/\log M)})$-factor away from the promised upper bound. Tweaking the algorithm to save the logarithmic factor uses the following lemma by [Heg95]. It shows that there exists an explicit ordering and a string s^i such that replacing $\mathrm{MAJ}(\mathcal{C}_i)$ in the basic algorithm leads to faster reduction of $|\mathcal{C}_i|$.

6.4.10. Lemma ([Heg95, Lemma 3.2]). *Let $L \in \mathbb{N}$ and $\mathcal{C} \subseteq \{0,1\}^L$. There exists $s \in \{0,1\}^L$ and permutation $\pi : [L] \to [L]$, such that for every $p \in [L]$, we have $|\mathcal{C}_p| \leq \frac{|\mathcal{C}|}{\max\{2,p\}}$, where $\mathcal{C}_p = \{c \in \mathcal{C} : c_{\{\pi(1),\ldots,\pi(p-1)\}} = s_{\{\pi(1),\ldots,\pi(p-1)\}},\ c_{\pi(p)} \neq s_{\pi(p)}\}$ is the set of strings in $\mathcal{C}$ that agree with s at $\pi(1),\ldots,\pi(p-1)$ and disagree at $\pi(p)$.*

We now describe the final algorithm.

1. Set $\mathcal{C}_1 := \mathcal{C}$, $N_1 := N$, and $c^1 := c$.
2. Repeat until $|\mathcal{C}_k| = 1$
   a. Let $s^k \in \{0,1\}^{N_k}$ be the string and $\pi^k : [N_k] \to [N_k]$ be the permutation obtained by applying Lemma 6.4.10 to $\mathcal{C}_k$ (with $L = N_k$)
   b. Search for the first (according to π^k) disagreement between s^k and c^k using the algorithm of Theorem 6.2.1. Suppose we find a disagreement at index $\pi^k(p_k) \in [N_k]$, i.e., s^k and c^k agree on the indices $I_k = \{\pi^k(1),\ldots,\pi^k(p_k-1)\}$
   c. Set $N_{k+1} := N_k - p_k$, $c^{k+1} := c^k_{[N_k]\setminus(I_k \cup \{\pi^k(p_k)\})}$ and $\mathcal{C}_{k+1} := \{u_{[N_k]\setminus(I_k\cup\{\pi^k(p_k)\})} : u \in \mathcal{C}_k,\ u_{I_k} = s^k_{I_k},\ u_{\pi^k(p_k)} \neq s^k_{\pi^k(p_k)}\}$
3. Output the unique element of $\mathcal{C}_k$.

Let r be the number of times the loop in Step 2 repeats and suppose in the k-th iteration we learned p_k bits of c. Then we have $\sum_{k=1}^{r} p_k \leq N$. The overall query complexity is $T = O(\sum_{k=1}^{r} \sqrt{p_k})$. Earlier we had $|\mathcal{C}_{k+1}| \leq |\mathcal{C}_k|/2$ and hence

[7] One has to be careful here because each run of the algorithm of Theorem 6.2.1 has a small error probability. Kothari shows how this can be handled *without* the super-constant blow-up in the overall complexity that would follow from naïve error reduction.

$r \leq O(\log M)$. But now, from Lemma 6.4.10 we have $|\mathcal{C}_{k+1}| \leq |\mathcal{C}_k|/\max\{2, p_k\}$. Since each iteration reduces the size of $\mathcal{C}_k$ by a factor of $\max\{2, p_k\}$, we have $\prod_{k=1}^{r} \max\{2, p_k\} \leq M$. Solving this optimization problem, i.e.,

$$\min\ T = \sum_{k=1}^{r} \sqrt{p_k} \quad \text{s.t.} \quad \prod_{k=1}^{r} \max\{2, p_k\} \leq M, \quad \sum_{k=1}^{r} p_k \leq N,$$

Kothari showed that

$$T = O(\sqrt{M}) \quad \text{if } M \leq N, \quad \text{and } T = O\Big(\sqrt{\frac{N \log M}{\log(N/\log M) + 1}}\Big) \quad \text{if } M > N.$$

□

Kothari [Kot14], improving upon [SG04, AS05], resolved a conjecture of Hunziker et al. [HMP+10] by showing the following upper bound for quantum query complexity of exact learning.

6.4.11. Theorem ([Kot14]). *For every concept class $\mathcal{C}$, there is a quantum exact learner for $\mathcal{C}$ using $O\Big(\sqrt{\frac{1/\gamma(\mathcal{C})}{\log(1/\gamma(\mathcal{C}))}} \log|\mathcal{C}|\Big)$ quantum membership queries.*

Proof sketch. The proof is very similar to the upper bound in Theorem 6.4.7, analyzed in terms of $\gamma(\mathcal{C})$ instead of (N, M). Consider the algorithm described in the proof of Theorem 6.4.7. In step (2b), learning an index at which s^k and c^k differ, reduces the size of C_k by a factor of $(1 - \gamma(C_k)) \leq (1 - \gamma(\mathcal{C}))$. If a disagreement in step (2b) was found at index $\pi^k(p_k)$ (i.e., c^k and s^k agree on the indices $I_k = \{\pi^k(1), \ldots, \pi^k(p_k - 1)\}$ and differ on $\pi^k(p_k)$), then this reduces the size of $\mathcal{C}_k$ by a factor at most $(1-\gamma(\mathcal{C}))^{p_k}$. Using this, we can now replace the constraint $\prod_{k=1}^{r} \max\{2, p_k\} \leq M$ by $M \prod_{k=1}^{r} (1 - \gamma(\mathcal{C})^{p_k} \geq 1$, since the target concept will remain in $\mathcal{C}$ after r rounds. It also easily follows that $\sum_k p_k \leq (\log M)/\gamma(\mathcal{C})$. Solving this new optimization problem

$$\min\ T = \sum_{k=1}^{r} \sqrt{p_k} \quad \text{s.t.} \quad M \prod_{k=1}^{r} (1 - \gamma(\mathcal{C}))^{p_k} \geq 1, \quad \sum_{k=1}^{r} p_k \leq \frac{\log M}{\gamma(\mathcal{C})},$$

Kothari showed that

$$T = O\Big(\sqrt{\frac{1/\gamma(\mathcal{C})}{\log(1/\gamma(\mathcal{C}))}} \log|\mathcal{C}|\Big).$$

□

Moshkin [Mos83] introduced another combinatorial parameter, which Hegedűs [Heg95] called the *extended teaching dimension* EXT-TD($\mathcal{C}$) of a concept class $\mathcal{C}$ (we shall not define EXT-TD($\mathcal{C}$) here, see [Heg95] for a precise definition). Building upon the work of [Mos83], Hegedűs proved the following theorem.

6.4.12. THEOREM ([Mos83],[Heg95, Theorem 3.1]). *Every classical exact learner for concept class $\mathcal{C}$ has to use $\Omega(\max\{\text{EXT-TD}(\mathcal{C}), \log|\mathcal{C}|\})$ membership queries. In the other direction, for every $\mathcal{C}$, there is a classical exact learner which learns $\mathcal{C}$ using $O\left(\frac{EXT\text{-}TD(\mathcal{C})}{\log(EXT\text{-}TD(\mathcal{C}))}\log|\mathcal{C}|\right)$ membership queries.*

Comparing this with Theorem 6.4.2, observe that both $1/\gamma(\mathcal{C})$ and EXT-TD($\mathcal{C}$) give lower bounds on classical query complexity, but the upper bound in terms of EXT-TD($\mathcal{C}$) is better by a logarithmic factor. Also for analyzing quantum complexity, EXT-TD($\mathcal{C}$) may be a superior parameter.

6.5 Results on sample complexity

6.5.1 Sample complexity of PAC learning

One of the most fundamental results in learning theory is that the sample complexity of $\mathcal{C}$ is tightly determined by a combinatorial parameter called the *VC dimension* of $\mathcal{C}$, named after Vapnik and Chervonenkis [VC71] and defined as follows.

6.5.1. DEFINITION. (VC dimension) Let $\mathcal{C}$ be a concept class over $\{0,1\}^n$. A set $\mathcal{S} = \{s_1, \ldots, s_t\} \subseteq \{0,1\}^n$ is said to be *shattered* by a concept class $\mathcal{C}$ if $\{(c(s_1)\cdots c(s_t)) : c \in \mathcal{C}\} = \{0,1\}^t$. In other words, for every labeling $\ell \in \{0,1\}^t$, there exists a $c \in \mathcal{C}$ such that $(c(s_1)\cdots c(s_t)) = \ell$. The *VC dimension* of $\mathcal{C}$ (denoted VC-dim($\mathcal{C}$)) is the size of a largest $\mathcal{S} \subseteq \{0,1\}^n$ that is shattered by $\mathcal{C}$.

In order to get a good intuition for this definition, in Figure 6.1 we consider two examples of concept classes $\mathcal{C} \subseteq \{0,1\}^2$ containing 9 concepts.

Blumer et al. [BEHW89] gave a surprising connection between the combinatorial parameter VC-dim($\mathcal{C}$) and sample complexity of PAC learning. They proved that the (ε,δ)-PAC sample complexity of a concept class $\mathcal{C}$ with VC dimension d, is lower bounded by $\Omega(d/\varepsilon + \log(1/\delta)/\varepsilon)$,[8] and they proved an upper bound that was worse by only a $\log(1/\varepsilon)$-factor. In recent work, Hanneke [Han16] (improving on Simon [Sim15]) got rid of this logarithmic factor,[9] showing that the lower bound of Blumer et al. is in fact optimal. Combining these bounds, we have the following theorem.

[8] It is not hard to see that the VC dimension of a concept class $\mathcal{C}$ also lower bounds the classical query complexity of exactly learning $\mathcal{C}$. By restricting to the concepts in the shattered set (for e.g., in Fig. 6.1, we could restrict $\{c_1, c_3, c_7, c_9\} \subseteq \mathcal{C}_1$ to the first two columns), an exact learning algorithm needs to make $\Omega(\text{VC-dim}(\mathcal{C}))$ many queries to identify an unknown concept with high probability.

[9] Hanneke's learner is not proper, meaning that its hypothesis h is not always in $\mathcal{C}$. It is still an open question whether the $\log(1/\varepsilon)$-factor can be removed for proper PAC learning. Our lower bounds in this chapter hold for all learners, quantum as well as classical, and proper as well as improper.

Concept class $\mathcal{C}_1$	Truth table			
c_1	0	1	0	1
c_2	0	1	1	0
c_3	1	0	0	1
c_4	1	0	1	0
c_5	1	1	0	1
c_6	0	1	1	1
c_7	0	0	1	1
c_8	0	1	0	0
c_9	1	1	1	1

Concept class $\mathcal{C}_2$	Truth table			
c_1	0	1	1	0
c_2	1	0	0	1
c_3	0	0	0	0
c_4	1	1	0	1
c_5	1	0	1	0
c_6	0	1	1	1
c_7	0	0	1	1
c_8	0	1	0	1
c_9	0	1	0	0

Figure 6.1: In the first table, the first two columns *contain* $\{0,1\}^2$, i.e., restricting the concepts $\{c_7, c_1, c_3, c_9\}$ to the first two columns we see all possible instantiations of $\{0,1\}^2$. This shows that VC-dim$(\mathcal{C}_1) \geq 2$ and observe that no three columns contain $\{0,1\}^3$, hence VC-dim$(\mathcal{C}_1) = 2$. In the second table the last three columns contain $\{0,1\}^3$, hence showing VC-dim$(\mathcal{C}_2) \geq 3$ and since there are only 9 concepts it is impossible to contain $\{0,1\}^4$ in any four columns.

6.5.2. Theorem ([BEHW89, Han16]). *Let $\mathcal{C}$ be a concept class that satisfies* VC-dim$(\mathcal{C}) = d+1$. *Then, $\Theta\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ examples are necessary and sufficient for an (ε,δ)-PAC learner for $\mathcal{C}$.*

This characterizes the number of samples necessary and sufficient for a classical PAC learning in terms of the VC dimension. How many *quantum* examples are needed to learn a concept class $\mathcal{C}$ of VC dimension d? Trivially, *upper* bounds on classical sample complexity imply upper bounds on quantum sample complexity. For some fixed distributions, in particular the uniform one, we will see in Section 6.7 that quantum examples can be more powerful than classical examples.

However, PAC learning requires a learner to be able to learn c under *all possible* distributions D, not just uniform. In Chapter 7, we show that quantum examples are *not more powerful* than classical examples in the PAC model, improving over the results of [AS05, Zha10].

6.5.3. Theorem ([AW17b]). *Let $\mathcal{C}$ be a concept class with* VC-dim$(\mathcal{C}) = d+1$. *Then, for every $\delta \in (0,1/2)$ and $\varepsilon \in (0,1/20)$, $\Omega\left(\frac{d}{\varepsilon} + \frac{1}{\varepsilon}\log\frac{1}{\delta}\right)$ examples are necessary for an (ε,δ)-quantum PAC learner for $\mathcal{C}$.*

The proof of this is fairly technical and we prove it in the next chapter. Using the upper bound from Theorem 6.5.2 and the lower bound above, it follows that classical and quantum sample complexity are equal up to constant factors for every concept class $\mathcal{C}$.

6.5.2 Sample complexity of agnostic learning

The following theorem characterizes the classical sample complexity of agnostic learning in terms of the VC dimension.

6.5.4. Theorem ([VC74, Sim96, Tal94]). *Let $\mathcal{C}$ be a concept class that satisfies* VC-dim$(\mathcal{C}) = d$. *Then,* $\Theta\left(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\right)$ *examples are necessary and sufficient for an (ε, δ)-agnostic learner for $\mathcal{C}$.*

The lower bound was proven by Vapnik and Chervonenkis [VC74] (see also Simon [Sim96]), and the upper bound was proven by Talagrand [Tal94]. Shalev-Shwartz and Ben-David [SB14, Section 6.4] call Theorems 6.5.2 and 6.5.4 the "Fundamental Theorem of PAC learning". It turns out that the quantum sample complexity of agnostic learning is equal (up to constant factors) to the classical sample complexity.

6.5.5. Theorem ([AW17b]). *Let $\mathcal{C}$ be a concept class with* VC-dim$(\mathcal{C}) = d$. *Then, for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/10)$,* $\Omega\left(\frac{d}{\varepsilon^2} + \frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$ *examples are necessary for an (ε, δ)-quantum agnostic learner for $\mathcal{C}$.*

The proof of the agnostic lower bound is similar to the proof Theorem 6.5.3 and we defer it to the next chapter.

We just saw that in sample complexity for the PAC and agnostic models, quantum examples do not provide an advantage. Gavinsky [Gav12] introduced a model of learning called "Predictive Quantum" (PQ), a variation of the quantum PAC model. He exhibited a *relational* concept class that is polynomial-time learnable in PQ, while any "reasonable" classical model requires an exponential number of labeled examples to learn the class.

6.6 The learnability of quantum states

Full-state tomography. So far, we considered learning concept classes of Boolean functions. In addition to learning *classical* objects, one may also consider the learnability of *quantum* objects. Here, a learner is given copies of an unknown quantum state ρ and the goal is to end up with a *classical description* of a quantum state σ that is in some sense *close* to ρ—and which sense of "closeness" we require makes a huge difference. Learning such a good approximation of ρ in trace distance is called *state tomography*.[10]

In general, an $(\log d)$-qubit state ρ is a Hermitian $d \times d$ matrix of trace 1, and hence described by roughly d^2 real parameters. Suppose we restrict attention to allowing only two-outcome measurements on the state ρ. Such a measurement is

[10] Trace norm of a matrix A is $\|A\|_{\mathrm{tr}} = \frac{1}{2}\sum_i |\lambda_i|$, where the λ_is are the eigenvalues of A.

specified by two positive semi-definite operators E and $\mathbf{1}_d - E$, and the probability for the measurement to yield the first outcome is $\mathrm{Tr}(E\rho)$. Since a two-outcome measurement gives at most one bit of information about ρ, $\widetilde{\Omega}(d^2)$ measurement results are necessary to learn a σ that is entry-wise very close to ρ. Recently, Haah et al. [HHJ+16] showed that this lower bound is true even for learning a σ that is close to ρ up to constant trace distance. Additionally, Haah et al. [HHJ+16] and O'Donnell and Wright [OW16] showed that $O(d^2/\varepsilon^2)$ copies of ρ are *sufficient* to produce a state σ with trace distance $\|\rho - \sigma\|_{\mathrm{tr}} \leq \varepsilon$.

Learning from measurements. Because of the exponential scaling in the number of qubits, the number of measurements needed for tomography of an arbitrary state on, say, 100 qubits is already prohibitively large.

Aaronson [Aar07] studied how well a quantum state ρ can be learned from measurement results, i.e., instead of being given ρ like in state tomography, we are given $\mathrm{Tr}(E_i\rho)$ where E_i is drawn from a known collection of 2-outcome measurements. In this setting, Aaronson showed an interesting and surprisingly efficient PAC-like result: from $O(n)$ measurement results, with measurements chosen i.i.d. according to an unknown distribution D on the set of all possible two-outcome measurements, we can construct an n-qubit quantum state σ that has roughly the same expectation value as ρ for "most" two-outcome measurements. In the latter, "most" is again measured under the same D that generated the measurements, just like in the usual PAC setting where the "approximate correctness" of the learner's hypothesis is evaluated under the same distribution D that generated the learner's examples. The output state σ can then be used to predict the behavior of ρ on two-outcome measurements, and it will give a good prediction for most measurements. Accordingly, $O(n)$ rather than $\exp(n)$ measurement results suffice for "pretty good tomography": to approximately learn an n-qubit state that is, maybe not close to ρ in trace distance, but still good enough for most practical purposes. More precisely, Aaronson's result is the following.

6.6.1. Theorem ([Aar07]). *For every $\delta, \varepsilon, \gamma > 0$, there exists a learner satisfying the following: for every distribution D on the set of two-outcome measurements, given $T = n \cdot \mathrm{poly}(1/\varepsilon, 1/\gamma, \log(1/\delta))$ measurement results $(E_1, b_1), \ldots, (E_T, b_T)$ where each E_i is drawn i.i.d. from D and b_i is a bit with $\Pr[b_i = 1] = \mathrm{Tr}(E_i\rho)$, with probability $\geq 1 - \delta$ the learner produces the classical description of a state σ such that*

$$\Pr_{E \sim D}\left[|\mathrm{Tr}(E\sigma) - \mathrm{Tr}(E\rho)| > \gamma\right] \leq \varepsilon.$$

Note that the "approximately correct" motivation of the original PAC model is now quantified by two parameters ε and γ, rather than only by one parameter ε as before: the output state σ is deemed approximately correct if the value $\mathrm{Tr}(E\sigma)$ has additive error at most γ (compared to the correct value $\mathrm{Tr}(E\rho)$), except with probability ε over the choice of E. We then want the output to be approximately

correct except with probability δ, like before. Note also that the theorem only says anything about the *sample* complexity of the learner (i.e., the number T of measurement results used to construct σ), not about the time complexity, which may be quite bad in general.

Proof sketch. The proof invokes general results due to Bartlett and Long [BL98] and Anthony and Bartlett [AB00] about learning classes of probabilistic functions[11] in terms of their γ-*fat-shattering dimension*. This generalizes VC dimension from Boolean to real-valued functions, as follows. For some set $\mathcal{E}$, let $\mathcal{C}$ be a class of functions $f : \mathcal{E} \to [0,1]$. We say that the set $S = \{E_1, \ldots, E_d\} \subseteq \mathcal{E}$ is γ-*fat-shattered* by $\mathcal{C}$ if there exist $\alpha_1, \ldots, \alpha_d \in [0,1]$ such that for all $Z \subseteq [d]$ there is an $f \in \mathcal{C}$ satisfying:

1. If $i \in Z$, then $f(E_i) \geq \alpha_i + \gamma$.
2. If $i \notin Z$, then $f(E_i) \leq \alpha_i - \gamma$.

The γ-*fat-shattering* dimension of $\mathcal{C}$ is the size of a largest S that is shattered by $\mathcal{C}$.[12]

For the application to learning quantum states, let $\mathcal{E}$ be the set of all n-qubit measurement operators. The relevant class of probabilistic functions corresponds to the n-qubit density matrices:

$$\mathcal{C} = \{f : \mathcal{E} \to [0,1] \mid \exists\ n\text{-qubit } \rho \text{ s.t. } \forall E \in \mathcal{E}, f(E) = \mathrm{Tr}(E\rho)\}.$$

Suppose the set $S = \{E_1, \ldots, E_d\}$ is γ-fat-shattered by $\mathcal{C}$. This means that for each string $z \in \{0,1\}^d$, there exists an n-qubit state ρ_z from which the bit z_i can be recovered using measurement E_i, with a γ-advantage over just outputting 1 with probability α_i. Such encodings $z \mapsto \rho_z$ of classical strings into quantum states are called *quantum random access codes*. Using known bounds on such codes [ANTV02], Aaronson shows that $d = O(n/\gamma^2)$. This upper bound on the γ-fat-shattering dimension of $\mathcal{C}$ can then be plugged into [AB00, BL98] to get the theorem. □

In a recent work, Aaronson [Aar17] considered the problem of *shadow tomography*, which we define informally now. Suppose ρ is an *unknown* d-dimensional quantum state and $\{E_1, \ldots, E_m\}$ is a collection of *known* 2-outcome measurements. As always, the learner needs to learn ρ, but instead of outputting a description of σ that is close to ρ like in full-state tomography, here the learner needs to output $\mathrm{Tr}(E_i\rho)$ up to additive error ε for every $i \in [m]$. Also, instead of obtaining measurement results (E_i, b_i) (like in Theorem 6.6.1), in shadow tomography the learner is given k copies of ρ (like in state tomography) and the goal

[11] A probabilistic function f over a set $\mathcal{S}$ is a function $f : \mathcal{S} \to [0,1]$.

[12] Note that if the functions in $\mathcal{C}$ have range $\{0,1\}$ and $\gamma > 0$, then this is just our usual VC dimension.

is to minimize k (as a function of $m, d, 1/\varepsilon$), the number of copies that suffice to perform shadow tomography.

Clearly, one way to solve shadow tomography is to ignore the measurements $\{E_1, \ldots, E_m\}$ completely and perform full-state tomography using copies of ρ. As we discussed earlier, it suffices to obtain $O(d^2/\varepsilon^2)$ copies [OW16, HHJ+16] of ρ in order to estimate ρ up to trace distance ε. Another possibility to solve shadow tomography is to use $O(1/\varepsilon^2)$ copies of ρ and estimate $\mathrm{Tr}(E_i\rho)$ up to additive precision ε for every $i \in [m]$. For this it suffices to obtain $O(m/\varepsilon^2)$ copies of ρ. Aaronson improved both these bounds *exponentially* by showing that shadow tomography can be solved using $k = \mathrm{poly}(\log d, \log m, 1/\varepsilon)$ copies of ρ. Although the proof of his result is fairly technical, we state his theorem and describe why there is an exponential savings in m, d below.

6.6.2. Theorem ([Aar17]). *For every $\varepsilon, \delta \in [0,1]$ and integers $m, d > 0$, given $k = \mathrm{poly}(\log m, \log d, 1/\varepsilon, \log(1/\delta))$ copies of an* unknown *d-dimensional quantum state ρ, and a* known *collection of 2-outcomes measurements $\{E_1, \ldots, E_m\}$, there exists a learner who, with probability at least $1-\delta$, outputs a set of numbers $b_1, \ldots, b_m \in [0,1]$ such that $|\mathrm{Tr}(E_i\rho) - b_i| \leq \varepsilon$ for every $i \in [m]$.*

Proof sketch. For simplicity, we let $\varepsilon = \delta = 1/3$. The proof is based on the idea of *postselected learning* introduced by Aaronson [Aar05a] in the context of one-way communication complexity.[13] Here, there are two parties Alice and Bob, who together want to solve a certain task. Suppose Alice is given a d-dimensional quantum state ρ (unknown to Bob) and they have common knowledge of a set of 2-outcome measurements $\{E_1, \ldots, E_m\}$. Alice needs to "describe" ρ to Bob in such a way that Bob can predict $\mathrm{Tr}(E_i\rho)$ (say up to additive error $1/3$) for every $i \in [m]$. Clearly an upper bound on the communication cost of this game is $\widetilde{O}(\min\{d^2, m\})$ bits, since Alice can either send the d^2 entries of ρ up to a certain precision or send estimates of $\mathrm{Tr}(E_i\rho)$ for every $i \in [m]$. Surprisingly, Aaronson [Aar05a] proposed a protocol involving only $\widetilde{O}(\min\{\log d, \log m\})$ qubits, which we sketch first.

Bob begins by simply "guessing" the state Alice possesses. Initially Bob assumes $\rho_0 = \mathbf{1}_d/d$, the maximally mixed state, and he keeps updating his guess. At the tth round, suppose Bob's current guess is ρ_t (whose classical description is also known to Alice), Alice helps Bob by telling him the index $j \in [m]$ on which $|\mathrm{Tr}(E_j\rho_t) - \mathrm{Tr}(E_j\rho)|$ is the largest and sends him an approximation of $b = \mathrm{Tr}(E_j\rho)$. Using this information, Bob updates $\rho_t \to \rho_{t+1}$ as follows: let $q = O(\log\log d)$ and F_t be a two-outcome measurement on $\rho_t^{\otimes q}$ that applies the POVM $\{E_j, \mathbf{1}_d - E_j\}$ (say, they correspond to $0, 1$ outcome respectively) to each of the q copies of ρ_t and accepts if and only if the number of 1-outcomes was at least $(b - 1/3)q$.[14]

[13] The "one-way communication" refers to the fact that communication is only allowed in one direction, i.e., Alice can send bits to Bob, not vice versa.

[14] We are implicitly assuming here that $b \geq 1/3$. There is an extra argument for this assumption which we skip here.

Suppose ρ'_{t+1} is the state obtained by postselecting on F_t accepting $\rho_t^{\otimes q}$, then ρ_{t+1} is the state obtained by tracing out the last $q-1$ registers of ρ'_{t+1}. Alice and Bob continue with this process for T rounds until Bob has a satisfactory ρ'. Aaronson showed that after $T = \widetilde{O}(\log d)$ rounds, with probability at least $2/3$, Bob will have ρ' which satisfies $|\mathrm{Tr}(E_i\rho) - \mathrm{Tr}(E_i\rho')| \leq 1/3$ for all $i \in [m]$.

We now get back to proving the theorem. In shadow tomography, there is no Alice, and Bob is replaced by a quantum learner. So, at the tth stage, without any assistance, the learner needs to figure out $j \in [m]$ for which $|\mathrm{Tr}(E_j\rho_t) - \mathrm{Tr}(E_j\rho)|$ is large! It is easy to decide if such a $j \in [m]$ exists using a variant of the Quantum OR lemma [HLM17]. To be precise, Aaronson used the Quantum OR lemma to show that, given $O(\log m)$ copies of ρ_t, there is a subroutine that outputs "yes" if there *exists* a $j \in [m]$ for which $|\mathrm{Tr}(E_j\rho_t) - \mathrm{Tr}(E_j\rho)| \geq 2/3$ and outputs "no" if $|\mathrm{Tr}(E_j\rho_t) - \mathrm{Tr}(E_j\rho)| \leq 1/3$ for every $i \in [m]$. However, the communication protocol in the previous paragraph crucially used that, in the "yes" instance of the subroutine, Bob knows j (not just the existence of j) in order to update $\rho_t \to \rho_{t+1}$. Aaronson fixes this by using a simple binary search over $\{E_1, \ldots, E_m\}$ to find such a j. Putting these ideas together, Aaronson shows that using $\mathrm{poly}(\log d, \log m)$ copies of ρ, the learner can solve the shadow tomography problem. □

Learning specific quantum states. Recently, there were a couple of works on learning the class of *stabilizer states*. Let $\mathcal{S}$ be a subgroup of the n-qubit Pauli group $\{\pm 1, \pm i\} \cdot \{\mathbf{1}_2, X, Y, Z\}^{\otimes n}$. A state $|\psi\rangle$ is said to be *stabilized* by $\mathcal{S}$, if for every $S \in \mathcal{S}$, $|\psi\rangle$ is a $+1$ eigenstate of S. A *stabilizer state* is defined as the *unique state* stabilized by a subgroup $\mathcal{S}$ of size $|\mathcal{S}| = 2^n$. Stabilizer states are interesting because, by the Gottesmann-Knill theorem [Got98], these states can be simulated efficiently on a classical computer.

Rocchetto [Roc17] considered whether the class of stabilizer states can be learnt in a PAC-like setting. Here, the learner is given examples of the form $(E_i, \mathrm{Tr}(E_i\rho))$, where $\rho = |\psi\rangle\langle\psi|$ is an unknown stabilizer state and E_i is a POVM element drawn from an unknown distribution over the set of two-outcome measurements. In this setting, Rocchetto showed that stabilizer states are *efficiently* learnable in both query and time complexity. Montanaro [Mon17a] considered another setting where the learner obtains copies of the unknown n-qubit stabilizer state $|\psi\rangle$ and goal is to identify $|\psi\rangle$. Montanaro constructed a quantum algorithm that identifies an unknown stabilizer state given n copies of the state and runs in time $O(n^3)$.

Another line of work, in a similar spirit of learning quantum objects, Cheng et al. [CHY16] studied how many states are sufficient to learn an unknown *quantum measurement*. Here the answer turns out to be linear in the *dimension* of the space, so exponential in the number of qubits. Learning an unknown quantum state becomes a *dual problem* to their question and using this connection they

can reprove the results of Aaronson [Aar07] in a different way.

6.7 Time complexity

In many ways, the best measure of efficient learning is low *time complexity*. While low sample complexity is a necessary condition for efficient learning, the information-theoretic sufficiency of a small sample is not much help in practice if *finding* a good hypothesis still takes much time.[15] In this section we describe a number of results where the best quantum learner has much lower time complexity than the best known classical learner.

6.7.1 Time-efficient quantum PAC learning

When trying to find examples of quantum speed-ups for learning, it makes sense to start with the most famous example of quantum speed-up we have: Shor's algorithm for factoring integers in polynomial time [Sho97]. It is widely assumed that classical computers cannot efficiently factor Blum integers (i.e., integers that are the product of two distinct primes of equal bit-length, each congruent to 3 mod 4).

Prior to Shor's discovery, Kearns and Valiant [KV94a] had already constructed a concept class $\mathcal{C}$ based on factoring, as an example of a simple and efficiently-representable concept class with small VC dimension that is not efficiently learnable. Roughly speaking, each concept $c \in \mathcal{C}$ corresponds to a Blum integer N, and a positively-labeled example for the concept reveals N. A concise description of c, however, depends on the factorization of N, which is assumed to be hard to compute by classical computers. Servedio and Gortler [SG04] observed that, thanks to Shor's algorithm, this class *is* efficiently PAC learnable by quantum computers. They similarly observed that the factoring-based concept class devised by Angluin and Kharitonov [AK95] to show hardness of learning even with membership queries, *is* easy to learn by quantum computers.

6.7.1. Theorem ([SG04]). *If there is no efficient classical algorithm for factoring Blum integers, then*

1. *there exists a concept class that is efficiently PAC learnable by quantum computers but not by classical computers;*

2. *there exists a concept class that is efficiently exactly learnable from membership queries by quantum computers but not by classical computers.*

[15] As is often the case: for many concept classes, finding a polynomial-sized hypothesis h that is consistent with a given set of examples is NP-hard.

One can construct classical one-way functions based on the assumption that factoring is hard. These functions can be broken (i.e., efficiently inverted) using quantum computers. However, there are other classical one-way functions that we do not known how to break with a quantum computer. Surprisingly, Servedio and Gortler [SG04] managed to construct concept classes with quantum-classical separation based on any classical one-way function—irrespective of whether that one-way function can be broken by a quantum computer! The construction builds concepts by combining instances of Simon's problem [Sim97] with the pseudorandom function family that one can obtain from the one-way function.

6.7.2. THEOREM ([SG04]). *If classical one-way functions exist, then there is a concept class $\mathcal{C}$ that is efficiently exactly learnable from membership queries by quantum computers but not by classical computers.*

6.7.2 Learning DNF from uniform quantum examples

As we saw in Section 6.3, Bshouty and Jackson [BJ99] introduced the model of learning from quantum examples. Their main positive result is to show that Disjunctive Normal Form (DNF) formulas[16] are learnable in polynomial time from quantum examples under the uniform distribution. For learning DNF under the uniform distribution from *classical* examples, the best upper bound is quasi-polynomial time [Ver90]. With the added power of *membership queries*, where the learner can actively ask for the label of any x of his choice, DNF formulas are known to be learnable in polynomial time under uniform D [Jac97], but polynomial-time learnability *without* membership queries is a longstanding open problem.

The classical polynomial-time algorithm for learning DNF using membership queries is Jackson's *harmonic sieve* algorithm [Jac97]. Roughly speaking it does the following. First, one can show that if the target concept $c : \{0,1\}^n \to \{0,1\}$ is an s-term DNF (i.e., a disjunction of at most s conjunctions of variables and negated variables) then there exists an n-bit parity function that agrees with c on a $1/2 + \Omega(1/s)$ fraction of the 2^n inputs. Moreover, the Goldreich-Levin algorithm [GL89] can be used to efficiently *find* such a parity function with the help of membership queries. This constitutes a "weak learner": an algorithm to find a hypothesis that agrees with the target concept with probability at least $1/2+1/\mathrm{poly}(s)$. Second, there are general techniques known as "boosting" [Fre95] that can convert a weak learner into a "strong" learner, i.e., one that produces a hypothesis that agrees with the target with probability $1-\varepsilon$ rather than probability $1/2+1/\mathrm{poly}(s)$. Typically such boosting algorithms assume access to a weak learner that can produce a weak hypothesis under every possible distribution D, rather than just uniform D. The idea is to start with distribution $D_1 = D$, and

[16] A formula is said to be a DNF if and only if it is a disjunction of conjunctions of one or more literals. An example of a DNF is $(x_1 \wedge x_2 \wedge \overline{x_4}) \vee (x_3 \wedge \overline{x_5} \wedge x_2) \vee (x_5 \wedge \overline{x_7} \wedge \overline{x_6})$.

use the weak learner to learn a weak hypothesis h_1 w.r.t. D_1. Then define a new distribution D_2 focusing on the inputs where the earlier hypothesis failed; use the weak learner to produce a weak hypothesis h_2 w.r.t. D_2, and so on. After $r = \text{poly}(s)$ such steps the overall hypothesis h is defined as a majority function applied to $(h_1, \ldots, h_r)$.[17] Note that when learning under fixed uniform D, we can only sample the first distribution $D_1 = D$ directly. Fortunately, if one looks at the subsequent distributions $D_2, D_3, \ldots, D_r$ produced by boosting in this particular case, sampling those distributions D_i can be efficiently "simulated" using samples from the uniform distribution. Putting these ideas together yields a classical polynomial-time learner for DNF under the uniform distribution, using membership queries. Jackson et al. [JTY02] showed how quantum membership queries can improve Jackson's classical algorithm for learning DNF with membership queries under the uniform distribution [Jac97].

The part of the classical harmonic sieve that uses membership queries is the Goldreich-Levin algorithm for finding a parity (i.e., a character function χ_S) that is a weak hypothesis. The key to the *quantum* learner is to observe that one can replace Goldreich-Levin by Fourier sampling from uniform quantum examples (see Section 6.2.2). Let $f = 1 - 2c$, which is just c in ± 1-notation. If χ_S has correlation $\Omega(1/s)$ with the target, then $\widehat{f}(S) = \Omega(1/s)$ and Fourier sampling outputs that S with probability $\Omega(1/s^2)$. Hence $\text{poly}(s)$ runs of Fourier sampling will with high probability give us a weak hypothesis. Because the state at step 3 of the Fourier sampling algorithm can be obtained with probability $1/2$ from a uniform quantum example, we do not require the use of membership queries anymore. Describing this algorithm (and the underlying classical harmonic sieve) in full detail is beyond the scope of this chapter, but the above sketch hopefully gives the main ideas of the result of [BJ99].

6.7.3. Theorem ([BJ99]). *The concept class of s-term DNF is efficiently PAC learnable under the uniform distribution from quantum examples.*

6.7.3 Learning linear functions and juntas from uniform quantum examples

Uniform quantum examples can be used for learning other things as well. For example, suppose $f(x) = a \cdot x \mod 2$ is a linear function over $\mathbb{F}_2$. Then the Fourier spectrum of f, viewed as a ± 1-valued function, has all its weight on χ_a. Hence by Fourier sampling we can perfectly recover a with $O(1)$ quantum sample complexity and $O(n)$ time complexity. In contrast, classical learners need $\Omega(n)$ examples to learn f, for the simple reason that each classical example (and even each membership query, if those are available to the learner too) gives at most one bit of information about the target concept.

[17] Note that this is not *proper* learning: the hypothesis h need not be an s-term DNF itself.

A more complicated and interesting example is learning functions that depend (possibly non-linearly) on at most k of the n input bits, with $k \ll n$. Such functions are called *k-juntas*, since they are "governed" by a small subset of the input bits. We want to learn such f up to error ε from uniform (quantum or classical) examples. A trivial learner would sample $O(2^k \log n)$ classical examples and then go over all $\binom{n}{k}$ possible sets of up k variables in order to find one that is consistent with the sample. This gives time complexity $O(n^k)$. The best known upper bound on time complexity [MOS04] is only slightly better: $O(n^{k\omega/(\omega+1)})$, where $\omega \in [2, 2.38]$ is the optimal exponent for matrix multiplication.

Time-efficiently learning k-juntas under the uniform distribution for $k = O(\log n)$ is a notorious bottleneck in classical learning theory, since it is a special case of DNF learning: every k-junta can be written as an s-term DNF with $s < 2^k$, by just taking the OR over the 1-inputs of the underlying k-bit function. In particular, if we want to efficiently learn poly(n)-term DNF from uniform examples (still an open problem, as mentioned in the previous section) then we should at least be able to efficiently learn $O(\log n)$-juntas (also still open).

Bshouty and Jackson's DNF learner from uniform quantum examples implies that we can learn k-juntas using poly$(2^k, n)$ quantum examples and time (for fixed ε, δ). Atıcı and Servedio [AS09] gave a more precise upper bound.

6.7.4. Theorem ([AS09]). *There exists a quantum learning algorithm for the class of k-juntas under the uniform distribution that uses $O(k \log(k)/\varepsilon)$ uniform quantum examples, $O(2^k)$ uniform classical examples, and time $O(nk \log(k)/\varepsilon + 2^k \log(1/\varepsilon))$ time.*

Proof sketch. The idea is to first use Fourier sampling from quantum examples to find the k variables (at least the ones with non-negligible influence), and then to use $O(2^k)$ uniform classical examples to learn (almost all of) the truth-table of the function on those variables.

View the target k-junta f as a function with range ± 1. Let the *influence* of variable x_i on f be

$$\mathrm{Inf}_i(f) = \sum_{S:S_i=1} \widehat{f}(S)^2 = \mathbb{E}_x\left[\left(\frac{f(x) - f(x \oplus e_i)}{2}\right)^2\right] = \Pr_x[f(x) \neq f(x \oplus e_i)],$$

where $x \oplus e_i$ is x after flipping its ith bit. If $S_i = 1$ for an i that is not in the junta, then $\widehat{f}(S) = 0$. Hence Fourier sampling returns an S such that $S_i = 1$ only for variables in the junta. $\mathrm{Inf}_i(f)$ is exactly the probability that $S_i = 1$. Hence for a fixed i, the probability that i does *not* appear in T Fourier samples is

$$(1 - \mathrm{Inf}_i(f))^T \leq e^{-T \mathrm{Inf}_i(f)}.$$

If we set $T = O(k \log(k)/\varepsilon)$ and let V be the union of the supports of the T Fourier samples, then with high probability V contains all junta variables except

those with $\mathrm{Inf}_i(f) \ll \varepsilon/k$ (the latter ones can be ignored since even their joint influence is negligible).

Now use $O(2^k \log(1/\varepsilon))$ uniform classical examples. With high probability, at least $1 - \varepsilon/2$ of all $2^{|V|}$ possible settings of the variables in V will appear, and we use those to formulate our hypothesis h (say with random values for the few inputs of the truth-table that we didn't see in our sample, and for the ones that appeared twice with inconsistent f-values). One can show that, with high probability, h will disagree with f on at most an ε-fraction of $\{0,1\}^n$. □

In a related result, Belovs [Bel13] gives a very tight analysis of the number of quantum membership queries (though not the time complexity) needed to exactly learn k-juntas whose underlying k-bit function is symmetric. For example, if the k-bit function is OR or Majority, then $O(k^{1/4})$ quantum membership queries suffice. For the case of Majority, $\Theta(k)$ classical membership queries are required, giving a fourth-power separation between quantum and classical membership query complexity of exact learning (see Corollary 6.4.4).

Learning k-Fourier-sparse functions. In work in progress [ACLW18], we consider exact learning of the concept class of k-Fourier-sparse Boolean functions

$$\mathcal{C} = \{c : \{0,1\}^n \to \{-1,1\} : |\mathrm{supp}(\widehat{c})| \leq k\}$$

using uniform examples. Note that the concept class of k-juntas studied by Atıcı and Servedio [AS09] is a special case of 2^k-Fourier-sparse Boolean functions. Learning Fourier-sparse Boolean functions has been studied for over a decade under the name of sparse recovery [HIKP12, APVZ14] having applications in compressed sensing and the data stream model. Classically, Haviv and Regev [HR16] showed that $\widetilde{\Theta}(nk)$ uniform examples of the form $(x, c(x))$ (where x sampled according to the uniform distribution on $\{0,1\}^n$) are necessary and sufficient to exactly learn $\mathcal{C}$.

In [ACLW18] we consider the setting where a quantum learner is given uniform quantum examples $\frac{1}{\sqrt{2^n}} \sum_x |x, c(x)\rangle$. We show that $O(k^2 \log k)$ quantum examples suffice to exactly learn an unknown concept in $\mathcal{C}$ and $\Omega(k \log k)$ quantum examples are necessary to learn $\mathcal{C}$ (importantly our bounds are independent of n). Our upper bound uses two observations. First given uniform quantum examples, a quantum learner can sample from the *Fourier distribution* $\{\widehat{c}(S)^2\}_{S \in \{0,1\}^n}$ (we discussed this in Section 6.3.2). Second, Gopalan et al. [GOS+11, Theorem 12] showed that the Fourier coefficients of a k-Fourier-sparse Boolean function $c : \{0,1\}^n \to \{-1,1\}$ are integer multiples of $2^{1-\lfloor \log k \rfloor}$. Putting these together, the probability to see every S (such that $\widehat{c}(S) \neq 0$) when sampling from the Fourier distribution is at least $1/k^2$. So if a quantum learner takes $O(k^2 \log k)$ samples from the Fourier distribution, then with high probability it would obtain the k non-zero Fourier coefficients of the unknown c. We conclude the proof by using

a similar argument as Haviv and Regev [HR16] to exactly learn the unknown c. We are still in the process of improving this upper bound. Our lower bound proof is similar to the information-theoretic proof in [AW17b] (a proof of which is presented in the next chapter).

6.8 Conclusion and future work

Quantum learning theory studies the theoretical aspects of quantum machine learning. In this chapter we surveyed what is known about this area. Specifically

- **Query complexity of exact learning.** The number of quantum membership queries needed to exactly learn a target concept can be polynomially smaller than the number of classical membership queries, but not much smaller than that.

- **Sample complexity.** For the distribution-independent models of PAC and agnostic learning, quantum examples give no significant advantage over classical random examples: for every concept class, the classical and quantum sample complexities are the same up to constant factors. In contrast, for some fixed distributions (e.g., uniform) quantum examples can be much better than classical examples.

- **Time complexity.** There exist concept classes that can be learned superpolynomially faster by quantum computers than by classical computers, for instance based on Shor's or Simon's algorithm. This holds both in the model of exact learning with membership queries, and in the model of PAC-learning. If one allows uniform quantum examples, DNF and juntas can be learned much more efficiently than we know how to do classically.

We end with a number of directions for future research.

- Bshouty and Jackson [BJ99] showed that DNF (i.e., disjunctions of conjunctions of variables and negations of variables) can be efficiently learned from uniform quantum examples. Is the same true of depth-3 circuits? And what about *constant-depth* circuits with unbounded fan-in AND/OR or even threshold gates, i.e., the concept classes AC^0 and TC^0—might even these be efficiently learnable from uniform quantum examples or even PAC-learnable? The latter is one of Scott Aaronson's "Ten Semi-Grand Challenges for Quantum Computing Theory" [Aar05b]. Classically, the best upper bounds on time complexity of learning AC^0 are quasi-polynomial under the uniform distribution [LMN93], and roughly $\exp(n^{1/3})$ in the PAC model (i.e., under all possible distributions) [KS04]; see [DS16] for a recent hardness result.

- Atıcı and Servedio [AS05] asked if for every $\mathcal{C}$, the upper bound in Corollary 6.4.4 can be improved to $D(\mathcal{C}) \leq O(nQ(\mathcal{C}) + Q(\mathcal{C})^2)$? Note that this bound is saturated for the concept class of $\mathcal{C} = \{e_i : i \in \{0, \ldots, N\}\}$ (where $D(\mathcal{C}) = \Omega(N)$ and $Q(\mathcal{C}) = O(\sqrt{N})$) and also for the concept class $\mathcal{C} = \{c(x) = a \cdot x : a \in \{0,1\}^n\}$ (where $D(\mathcal{C}) = \Omega(n)$ and $Q(\mathcal{C}) = O(1)$).

- Can we learn k-Fourier-sparse Boolean functions using $O(k \log k)$ uniform quantum examples?

- Can we characterize the classical and quantum query complexity of exactly learning a concept class $\mathcal{C}$ in terms of the combinatorial parameter $\gamma(\mathcal{C})$, or in terms of the extended teaching dimension of $\mathcal{C}$?

- Can we find more instances of concept classes where quantum examples are beneficial when learning w.r.t. some fixed distribution (uniform or otherwise), or some restricted set of distributions?

- Can we find examples of quantum speed-up in Angluin's [Ang87] model of equivalence queries plus membership queries?

- Most research in quantum learning theory has focused on concept classes of Boolean functions. What about learning classes of *real-valued* or even *vector-valued* functions?

- Can we find a *proper* quantum PAC learner with optimal sample complexity, i.e., one whose output hypothesis lies in $\mathcal{C}$ itself? Or a *proper* efficient quantum learner for DNF using uniform quantum examples?

- Can we find practical machine learning problems with a large provable quantum speed-up?

- Can we use quantum machine learning for "quantum supremacy", i.e., for solving some task using 50–100 qubits in a way that is convincingly faster than possible on large classical computers? (See for example [AC17] for some complexity results concerning quantum supremacy.)

Chapter 7

Quantum sample complexity

This chapter is based on the paper "Optimal Quantum Sample Complexity of Learning Algorithms", by S. Arunachalam and R. de Wolf [AW17b].

Abstract. In the previous chapter we discussed the PAC and agnostic learning models and saw that the VC dimension of a concept class $\mathcal{C}$ captures the number of classical examples needed to learn an unknown target concept in these models. Specifically, in the classical PAC model $\Theta\left(\frac{d}{\varepsilon}+\frac{\log(1/\delta)}{\varepsilon}\right)$ examples are necessary and sufficient for a learner to output, with probability $1-\delta$, an hypothesis h that is ε-close to the target concept c (measured under D). In the related classical *agnostic* model, where the samples need not come from a $c \in \mathcal{C}$, we know that $\Theta\left(\frac{d}{\varepsilon^2}+\frac{\log(1/\delta)}{\varepsilon^2}\right)$ examples are necessary and sufficient to output an hypothesis $h \in \mathcal{C}$ whose error is at most ε worse than the error of the best concept in $\mathcal{C}$.

In this chapter, we will analyze *quantum* sample complexity. We will prove Theorem 6.5.3 (showing an $\Omega\left(\frac{d}{\varepsilon}+\frac{\log(1/\delta)}{\varepsilon}\right)$ lower bound for PAC quantum sample complexity) and Theorem 6.5.5 (showing an $\Omega\left(\frac{d}{\varepsilon^2}+\frac{\log(1/\delta)}{\varepsilon^2}\right)$ lower bound for agnostic quantum sample complexity). Along with the classical upper bound, this shows that quantum and classical sample complexity are in fact equal up to constant factors in both the PAC and agnostic models.

Contents

7.1 Sample complexity and VC dimension

In this section, we quickly recap from the previous chapter, the PAC model of learning and the agnostic model of learning before describing our results.

7.1.1 The PAC setting

Leslie Valiant's Probably Approximately Correct (PAC) model [Val84] gives a precise complexity-theoretic definition of what it means for a concept class to be (efficiently) learnable. Let $\mathcal{C} \subseteq \{f : \{0,1\}^n \to \{0,1\}\}$ be a concept class. The goal of a learning algorithm (the learner) is to probably approximate some unknown *target concept* $c \in \mathcal{C}$ from random *labeled examples* of the form $(x, c(x))$ where x is distributed according to some unknown distribution D over $\{0,1\}^n$. After processing a number of such examples (hopefully not too many), the learner outputs some *hypothesis* h. We say that h is *ε-approximately correct* (w.r.t. the

target concept c) if its error probability under D is at most ε: $\Pr_{x\sim D}[h(x) \neq c(x)] \leq \varepsilon$. Note that the learning phase and the evaluation phase (i.e., whether a hypothesis is approximately correct) are according to the same distribution D—as if the learner is taught and then tested by the same teacher. An (ε, δ)-learner for the concept class $\mathcal{C}$ is one whose hypothesis is probably approximately correct:

> For all target concepts $c \in \mathcal{C}$ and distributions D:
> Pr[the learner's output h is ε-approximately correct] $\geq 1 - \delta$,

where the probability is over the sequence of examples and the learner's internal randomness. Of course, we want the learner to be as efficient as possible. Its *sample complexity* is the worst-case number of examples it uses, and its *time complexity* is the worst-case running time of the learner. In this chapter, we focus on sample complexity. This allows us to ignore technical issues of how the runtime of an algorithm is measured, and in what form the hypothesis h is given as output by the learner.

The sample complexity of a concept class $\mathcal{C}$ is the sample complexity of the most efficient learner for $\mathcal{C}$. It is a function of ε, δ, and of course of $\mathcal{C}$ itself. One of the most fundamental results in learning theory is that the sample complexity of $\mathcal{C}$ is tightly determined by the VC dimension of $\mathcal{C}$. Knowing this VC dimension (and ε, δ) already tells us the sample complexity of $\mathcal{C}$ up to constant factors. Blumer et al. [BEHW89] proved that the sample complexity of $\mathcal{C}$ is lower bounded by $\Omega(d/\varepsilon + \log(1/\delta)/\varepsilon)$ and in a very recent work, Hanneke [Han16] (improving on Simon [Sim15]) showed that the lower bound of Blumer et al. is in fact optimal: the sample complexity of $\mathcal{C}$ in the PAC setting is

$$\Theta\Big(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\Big). \tag{7.1}$$

7.1.2 The agnostic setting

The PAC model assumes that the labeled examples are generated according to a target concept $c \in \mathcal{C}$. However, in many learning situations that is not a realistic assumption, for example when the examples are noisy in some way or when we have no reason to believe there is an underlying target concept at all. The *agnostic* model of learning, introduced by Haussler [Hau92] and Kearns et al. [KSS94], takes this into account. Here, the examples are generated according to a distribution D on $\{0,1\}^{n+1}$. The error of a specific concept $c : \{0,1\}^n \to \{0,1\}$ is defined to be $\mathrm{err}_D(c) = \Pr_{(x,b)\sim D}[c(x) \neq b]$. When we are restricted to hypotheses in $\mathcal{C}$, we would like to find the hypothesis that minimizes $\mathrm{err}_D(c)$ over all $c \in \mathcal{C}$. However, it may require very many examples to do that exactly. In the spirit of the PAC model, the goal of the learner is now to output an $h \in \mathcal{C}$ whose error is at most an additive ε worse than that of the best (= lowest-error) concepts in $\mathcal{C}$.

Like in the PAC model, the optimal sample complexity of such agnostic learners is tightly determined by the VC dimension of $\mathcal{C}$: it is

$$\Theta\Big(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\Big), \tag{7.2}$$

where the lower bound was proven by Vapnik and Chervonenkis [VC74] (see also Simon [Sim96]), and the upper bound was proven by Talagrand [Tal94]. Shalev-Shwartz and Ben-David [SB14, Section 6.4] call Eq. (7.1) and Eq. (7.2) the "Fundamental Theorem of PAC learning."

Organization. For an introduction to the classical and quantum learning models, see Chapter 6. In Section 7.2, we formally state our results and give proof sketches of our bounds. In Section 7.3, we present some preliminaries in quantum information theory, state some important facts that we use often and describe the pretty good measurement. In Section 7.4 we prove our information-theoretic lower bounds both for classical and quantum learning. In Section 7.5 we prove an optimal quantum lower bound for PAC and agnostic learning by viewing the learning process as a state identification problem. In Section 7.6 we mention two additional results on learning under classification noise and distinguishing codeword states. We conclude in Section 7.7 with some open questions for further work.

7.2 Our results

In this chapter we are interested in *quantum* sample complexity. Here a *quantum example* for some concept $c : \{0,1\}^n \to \{0,1\}$, according to some distribution D, corresponds to an $(n+1)$-qubit state

$$\sum_{x \in \{0,1\}^n} \sqrt{D(x)}|x, c(x)\rangle.$$

How many quantum examples are needed to learn a concept class $\mathcal{C}$ of VC dimension d? Since a learner can just measure a quantum example in order to obtain a classical example, the *upper* bounds on classical sample complexity trivially imply the same upper bounds on quantum sample complexity. But what about the lower bounds? Are there situations where quantum examples are more powerful than classical? Indeed there are. In the previous chapter, we already mentioned the results of Bshouty and Jackson [BJ99] for learning DNF under the uniform distribution without membership queries. Another good example is the learnability of the concept class of linear functions over $\mathbb{F}_2$, $\mathcal{C} = \{c(x) = a \cdot x : a \in \{0,1\}^n\}$, again under the uniform distribution D using the Bernstein-Vazirani algorithm [BV97].

Atıcı and Servedio [AS09] considered learning *k-juntas* from quantum examples under the uniform distribution. However, PAC learning requires a learner to learn c under *all possible* distributions D, not just the uniform one. The success probability of the Bernstein-Vazirani algorithm deteriorates sharply when D is far from uniform, but that does not rule out the existence of other quantum learners that use $o(n)$ quantum examples and succeed for all D.

Our main result in this chapter is that quantum examples are not actually more powerful than classical labeled examples in the PAC model and in the agnostic model: we prove that the lower bounds on classical sample complexity of Eq. (7.1) and Eq. (7.2) hold for quantum examples as well. Accordingly, despite several distribution-specific speed-ups, quantum examples do not significantly reduce sample complexity if we require our learner to work for all distributions D. This should be contrasted with the situation when considering the *time complexity* of learning, where Servedio and Gortler [SG04] considered a concept class that can be PAC-learned in polynomial time by a quantum computer, even with only classical examples, but that cannot be PAC-learned in polynomial time by a classical learner unless Blum integers can be factored in polynomial time (which is widely believed to be false).

Earlier work on quantum sample complexity had already gotten close to extending the lower bound of Eq. (7.1) to PAC learning from quantum examples. Atıcı and Servedio [AS05] first proved a lower bound of $\Omega(\sqrt{d}/\varepsilon + d + \log(1/\delta)/\varepsilon)$ using the so-called "hybrid method." Their proof technique was subsequently pushed further by Zhang [Zha10] to

$$\Omega\Big(\frac{d^{1-\eta}}{\varepsilon} + d + \frac{\log(1/\delta)}{\varepsilon}\Big) \text{ for arbitrarily small constant } \eta > 0. \tag{7.3}$$

Here we optimize these bounds, removing the η and achieving the optimal lower bound for quantum sample complexity in the PAC model (Eq. (7.1)).

We also show that the lower bound (Eq. (7.2)) for the agnostic model extends to quantum examples. As far as we know, in contrast to the PAC model, no earlier results were known for quantum sample complexity in the agnostic model.

We have two different proof approaches, which we sketch below.

7.2.1 Proof sketch: An information-theoretic argument

In Section 7.4 we give a fairly intuitive information-theoretic argument that gives optimal lower bounds for classical sample complexity, and that gives nearly-optimal lower bounds for quantum sample complexity. Let us first see how we can prove the classical PAC lower bound of Eq. (7.1). Suppose $\mathcal{S} = \{s_0, s_1, \ldots, s_d\}$ is shattered by $\mathcal{C}$ (we now assume VC dimension $d+1$ for ease of notation). Then we can consider a distribution D that puts probability $1 - 4\varepsilon$ on s_0 and proba-

bility $4\varepsilon/d$ on each of $s_1, \ldots, s_d$.[1] For every possible labeling $(\ell_1 \ldots \ell_d) \in \{0,1\}^d$ of $s_1, \ldots, s_d$ there will be a concept $c \in \mathcal{C}$ that labels s_0 with 0, and labels s_i with ℓ_i for all $i \in \{1, \ldots, d\}$. Under D, most examples will be $(s_0, 0)$ and hence give us no information when we are learning one of those 2^d concepts. Suppose we have a learner that ε-approximates c with high probability under this D using T examples. Informally, our information-theoretic argument has the following three steps:

1. In order to ε-approximate c, the learner has to learn the c-labels of at least $3/4$ of the $s_1, \ldots, s_d$ (since together these have 4ε of the D-weight, and we want an ε-approximation). As all 2^d labelings are possible, the T examples together contain $\Omega(d)$ bits of information about c.

2. T examples give at most T times as much information about c as one example.

3. One example gives only $O(\varepsilon)$ bits of information about c, because it will tell us one of the labels of $s_1, \ldots, s_d$ only with probability 4ε (and otherwise it just gives $c(s_0) = 0$).

Putting these steps together implies $T = \Omega(d/\varepsilon)$.[2] This argument for the PAC setting is similar to an algorithmic-information argument of Apolloni and Gentile [AG98] and an information-theoretic argument for variants of the PAC model with noisy examples of Gentile and Helmbold [GH01].

As far as we know, this type of reasoning has not yet been applied to the sample complexity of *agnostic* learning. To get good lower bounds there, we consider a set of distributions D_a, indexed by d-bit string a. These distributions still have the property that if a learner gets ε-close to the minimal error, then it will have to learn $\Omega(d)$ bits of information about the distribution (i.e., about a). Hence the first step of the argument remains the same. The second step of our argument also remains the same, and the third step shows an upper bound of $O(\varepsilon^2)$ on the amount of information that the learner can get from one example. This then implies $T = \Omega(d/\varepsilon^2)$. We can also reformulate this for the case where we want the *expected* additional error of the hypothesis over the best classifier in $\mathcal{C}$ to be at most ε, which is how lower bounds are often stated in learning theory. We emphasize that our information-theoretic proof is simpler than the proofs in [AB09, Aud09, SB14, KP16].

This information-theoretic approach recovers the optimal classical bounds on sample complexity, but also generalizes readily to the quantum case where the

[1] We remark that the distributions used here for proving lower bounds on quantum sample complexity have been used by Ehrenfeucht et al. [EHKV89] for analyzing classical PAC sample complexity.

[2] The other part of the lower bound of Eq. (7.1) does not depend on d and is fairly easy to prove.

learner gets T quantum examples. To obtain lower bounds on quantum sample complexity we use the same distributions D (now corresponding to a coherent quantum state) and basically just need to re-analyze the third step of the argument. In the PAC setting we show that one quantum example gives at most $O(\varepsilon \log(d/\varepsilon))$ bits of information about c, and in the agnostic setting it gives $O(\varepsilon^2 \log(d/\varepsilon))$ bits. This implies lower bounds on sample complexity that are only a logarithmic factor worse than the optimal classical bounds for the PAC setting (Eq. (7.1)) and the agnostic setting (Eq. (7.2)). This is not quite optimal yet, but already better than the previous best known lower bound (Eq. (7.3)). The logarithmic loss in step 3 is actually inherent in this information-theoretic argument: in some cases a quantum example *can* give roughly $\varepsilon \log d$ bits of information about c, for example when c comes from the concept class of linear functions.

7.2.2 Proof sketch: A state-identification argument

In order to get rid of the logarithmic factor we then try another proof approach, which views learning from quantum examples as a quantum state identification problem: we are given T copies of the quantum example for some concept c and need to ε-approximate c from this. In order to render ε-approximation of c equivalent to exact identification of c, we use good linear error-correcting codes, restricting to concepts whose d-bit labeling of the elements of the shattered set $s_1, \ldots, s_d$ corresponds to a codeword. We then have $2^{\Omega(d)}$ possible concepts, one for each codeword, and need to identify the target concept from a quantum state that is the tensor product of T identical quantum examples.

State-identification problems have been well studied, and many tools are available for analyzing them. In particular, we will use the so-called "Pretty Good Measurement" (PGM, also known as "square root measurement" [HJS+96]) introduced by Hausladen and Wootters [HW94]. The PGM is a specific measurement that one can always use for state identification, and whose success probability is no more than quadratically worse than that of the very best measurement.[3] In Section 7.5 we use Fourier analysis to give an exact analysis of the average success probability of the PGM on the state-identification problems that come from both the PAC and the agnostic model. This analysis could be useful in other settings as well. Here it implies that the number of quantum examples, T, is lower bounded by Eq. (7.1) in the PAC setting, and by Eq. (7.2) in the agnostic setting.

Using the Pretty Good Measurement, we are also able to prove lower bounds for PAC learning under *random classification noise*, which models the real-world situation that the learning data can have some errors. Classically in the random classification noise model (introduced by Angluin and Laird [AL88]), instead of

[3] Even better, in our application the PGM *is* the optimal measurement, though this is not essential for our proof.

obtaining labeled examples $(x, c(x))$ for some unknown $c \in \mathcal{C}$, the learner obtains *noisy examples* (x, b_x), where $b_x = c(x)$ with probability $1-\eta$ and $b_x = 1-c(x)$ with probability η, for some *noise rate* $\eta \in [0, 1/2)$. Similarly, in the quantum learning model we could naturally define a *noisy quantum example* as an $(n+1)$-qubit state

$$\sum_{x\in\{0,1\}^n} \sqrt{(1-\eta)D(x)}|x, c(x)\rangle + \sqrt{\eta D(x)}|x, 1-c(x)\rangle.$$

Using the PGM, we are able to show that the quantum sample complexity of PAC learning a concept class $\mathcal{C}$ under random classification noise is:

$$\Omega\Big(\frac{d}{(1-2\eta)^2\varepsilon} + \frac{\log(1/\delta)}{(1-2\eta)^2\varepsilon}\Big). \tag{7.4}$$

We remark here that the best known classical sample complexity lower bound (see [Sim96]) under the random classification noise is equal to the quantum sample complexity lower bound proven in Eq. (7.4).

Related work. The use of Fourier analysis in analyzing the success probability of the Pretty Good Measurement in quantum state identification appears in a number of earlier works. By considering the dihedral hidden subgroup problem (DHSP) as a state identification problem, Bacon et al. [BCD06] show that the PGM is the optimal measurement for DHSP and prove a lower bound on the sample complexity of $\Omega(\log|\mathscr{G}|)$ for a dihedral group $\mathscr{G}$ using Fourier analysis. Ambainis and Montanaro [AM14] view the "search with wildcard" problem as a state identification problem. Using ideas similar to ours, they show that the (x, y)-th entry of the Gram matrix for the ensemble depends on the Hamming distance between x and y, allowing them to use Fourier analysis to obtain an upper bound on the success probability of the state identification problem using the PGM.

7.3 Preliminaries

7.3.1 Quantum information theory

We will introduce the basics of information theory here, referring to [CT91] for more on classical information theory and [NC02, Wat11] for more on quantum information theory.

We denote random variables in bold, such as $\mathbf{A}, \mathbf{B}$. For a probability vector $(p_1, \ldots, p_k)$ (where $\sum_{i\in[k]} p_i = 1$), the entropy function is defined as

$$H(p_1, \ldots, p_k) = -\sum_{i\in[k]} p_i \log p_i.$$

When $k = 2$, with $p_1 = p$ and $p_2 = 1 - p$, we denote the binary entropy function as $H(p)$. For a state ρ_{AB} on the Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$, we let ρ_A be the reduced state after taking the partial trace over $\mathcal{H}_B$. The entropy of a quantum state ρ_A is defined as $S(\mathbf{A}) = -\mathrm{Tr}(\rho_A \log \rho_A)$. The mutual information is defined as $I(\mathbf{A} : \mathbf{B}) = S(\mathbf{A}) + S(\mathbf{B}) - S(\mathbf{AB})$, and conditional entropy is defined as $S(\mathbf{A}|\mathbf{B}) = S(\mathbf{AB}) - S(\mathbf{B})$. Classical information-theoretic quantities correspond to the special case where ρ is a diagonal matrix whose diagonal corresponds to the probability distribution of the random variable. Writing ρ_A in its eigenbasis, it follows that $S(\mathbf{A}) = H(\lambda_1, \ldots, \lambda_{\dim(\rho_A)})$, where $\lambda_1, \ldots, \lambda_{\dim(\rho_A)}$ are the eigenvalues of ρ. If ρ_A is a pure state, $S(\mathbf{A}) = 0$.

7.3.2 The pretty good measurement

Consider an ensemble of d-dimensional quantum states, $\mathcal{E} = \{(p_i, |\psi_i\rangle)\}_{i\in[m]}$, where $\sum_{i\in[m]} p_i = 1$. Suppose we are given an unknown state $|\psi_i\rangle$ sampled according to the probabilities and we are interested in maximizing the average probability of success to identify the state that we are given. For a POVM specified by positive semidefinite matrices $\mathcal{M} = \{M_i\}_{i\in[m]}$, the probability of obtaining outcome j equals $\langle\psi_i|M_j|\psi_i\rangle$. The average success probability is defined as

$$P_{\mathcal{M}}(\mathcal{E}) = \sum_{i=1}^{m} p_i \langle\psi_i|M_i|\psi_i\rangle.$$

Let $P^{opt}(\mathcal{E}) = \max_{\mathcal{M}} P_{\mathcal{M}}(\mathcal{E})$ denote the optimal average success probability of $\mathcal{E}$, where the maximization is over the set of valid m-outcome POVMs.

For every ensemble $\mathcal{E}$, the so-called *Pretty Good Measurement* (PGM) is a specific POVM (depending on the ensemble $\mathcal{E}$) that does *reasonably* well against $\mathcal{E}$. The PGM is defined as follows: let $|\psi'_i\rangle = \sqrt{p_i}|\psi_i\rangle$, and $\mathcal{E}' = \{|\psi'_i\rangle : i \in [m]\}$ be the set of states in $\mathcal{E}$, renormalized to reflect their probabilities. Define $\rho = \sum_{i\in[m]} |\psi'_i\rangle\langle\psi'_i|$. The PGM is the set of measurement operators $\{|\nu_i\rangle\langle\nu_i|\}_{i\in[m]}$ where $|\nu_i\rangle = \rho^{-1/2}|\psi'_i\rangle$ (the inverse square root of ρ is taken over its non-zero eigenvalues). It is not hard to verify this is a valid POVM:

$$\sum_{i=1}^{m} |\nu_i\rangle\langle\nu_i| = \rho^{-1/2} \left(\sum_{i=1}^{m} |\psi'_i\rangle\langle\psi'_i| \right) \rho^{-1/2} = \mathbf{1}_d.$$

Suppose $P^{PGM}(\mathcal{E})$ is defined as the average success probability of identifying the states in $\mathcal{E}$ using the PGM, then clearly $P^{PGM}(\mathcal{E}) \leq P^{opt}(\mathcal{E})$ (because $P^{opt}(\mathcal{E})$ is a maximization over all valid POVMs). Barnum and Knill [BK02] furthermore proved that $P^{opt}(\mathcal{E})$ can be at most quadratically lesser than $P^{PGM}(\mathcal{E})$.

7.3.1. Theorem (Barnum and Knill [BK02]). *Let $\mathcal{E} = \{(p_i, |\psi_i\rangle)\}_{i\in[m]}$ be an ensemble of d-dimensional quantum states, where $\sum_{i\in[m]} p_i = 1$. Let $P^{opt}(\mathcal{E})$ and*

$P^{PGM}(\mathcal{E})$ denote the average success probability of identifying the states in $\mathcal{E}$ using the optimal m-outcome POVM and the PGM respectively. Then,

$$P^{opt}(\mathcal{E})^2 \leq P^{PGM}(\mathcal{E}) \leq P^{opt}(\mathcal{E}).$$

Proof. The upper bound on $P^{PGM}(\mathcal{E})$ is trivial. For completeness we give a simple proof of $P^{opt}(\mathcal{E})^2 \leq P^{PGM}(\mathcal{E})$ below (similar to [Mon07]). Let G be the Gram matrix for the set $\mathcal{E}'$, i.e., $G(i,j) = \langle\psi'_i|\psi'_j\rangle$ for $i,j \in [m]$. It can be verified that $\sqrt{G}(i,j) = \langle\psi'_i|\rho^{-1/2}|\psi'_j\rangle$. Hence

$$\begin{aligned} P^{PGM}(\mathcal{E}) = \sum_{i\in[m]} p_i |\langle\nu_i|\psi_i\rangle|^2 &= \sum_{i\in[m]} |\langle\nu_i|\psi'_i\rangle|^2 \\ &= \sum_{i\in[m]} \langle\psi'_i|\rho^{-1/2}|\psi'_i\rangle^2 = \sum_{i\in[m]} \sqrt{G}(i,i)^2. \end{aligned}$$

Suppose $\mathcal{M}$ is the optimal measurement that maximizes $P^{opt}(\mathcal{E})$. Since $\mathcal{E}$ consists of pure states, by a result of Eldar et al. [EMV03], we can assume without loss of generality that the measurement operators in $\mathcal{M}$ are rank-1, so $M_i = |\mu_i\rangle\langle\mu_i|$ for some $|\mu_i\rangle$. Note that

$$\begin{aligned} \sum_{i\in[m]} \langle\mu_i|\rho^{1/2}|\mu_i\rangle^2 &\leq \sum_{i,j\in[m]} |\langle\mu_i|\rho^{1/2}|\mu_j\rangle|^2 \\ &= \mathrm{Tr}\Big(\sum_{i\in[m]} |\mu_i\rangle\langle\mu_i|\rho^{1/2} \sum_{j\in[m]} |\mu_j\rangle\langle\mu_j|\rho^{1/2}\Big) = \mathrm{Tr}(\rho) = 1 \end{aligned} \tag{7.5}$$

Then, using the Cauchy-Schwarz inequality, we have

$$\begin{aligned} P^{opt}(\mathcal{E}) = \sum_{i\in[m]} |\langle\mu_i|\psi'_i\rangle|^2 &= \sum_{i\in[m]} |\langle\mu_i|\rho^{1/4}\rho^{-1/4}|\psi'_i\rangle|^2 \\ &\leq \sum_{i\in[m]} \langle\mu_i|\rho^{1/2}|\mu_i\rangle\langle\psi'_i|\rho^{-1/2}|\psi'_i\rangle \\ &\leq \sqrt{\sum_{i\in[m]} \langle\mu_i|\rho^{1/2}|\mu_i\rangle^2} \sqrt{\sum_{i\in[m]} \langle\psi'_i|\rho^{-1/2}|\psi'_i\rangle^2} \\ &\leq \sqrt{\sum_{i\in[m]} \langle\psi'_i|\rho^{-1/2}|\psi'_i\rangle^2} \\ &= \sqrt{P^{PGM}(\mathcal{E})}, \end{aligned}$$

where the last inequality used Eq. (7.5). □

The above shows that for all ensembles $\mathcal{E}$, the PGM for that ensemble is not much worse than the optimal measurement. In some cases the PGM *is*

the optimal measurement. In particular, an ensemble $\mathcal{E}$ is called *geometrically uniform* if $\mathcal{E} = \{U_i|\varphi\rangle : i \in [m]\}$ for some Abelian group[4] of matrices $\{U_i\}_{i\in[m]}$ and state $|\varphi\rangle$. Eldar and Forney [EF01] showed $P^{opt}(\mathcal{E}) = P^{PGM}(\mathcal{E})$ for such $\mathcal{E}$.

7.3.3 Known results and required claims

The following theorems characterize the sample complexity of classical PAC and agnostic learning.

7.3.2. Theorem ([BEHW89, Han16]). *Let $\mathcal{C}$ be a concept class that satisfies* VC-dim$(\mathcal{C}) = d + 1$. *Then, $\Theta\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ examples are necessary and sufficient for an (ε, δ)-PAC learner for $\mathcal{C}$.*

7.3.3. Theorem ([VC74, Sim96, Tal94]). *Let $\mathcal{C}$ be a concept class that satisfies* VC-dim$(\mathcal{C}) = d$. *Then, $\Theta\left(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\right)$ examples are necessary and sufficient for an (ε, δ)-agnostic learner for $\mathcal{C}$.*

We will use the following theorem from the theory of error-correcting codes:

7.3.4. Theorem. *For every sufficiently large integer n, there exists an integer $k \in [n/4, n]$ and a matrix $M \in \mathbb{F}_2^{n\times k}$ of rank k, such that the associated $[n, k, d]_2$ linear code $\{Mx : x \in \{0,1\}^k\}$ has minimal distance $d \geq n/8$.*

We will need the following claims later

7.3.5. Claim. *Let $f : \{0,1\}^m \to \mathbb{R}$ and let $M \in \mathbb{F}_2^{m\times k}$. Then the Fourier coefficients of $f \circ M$ are $\widehat{f \circ M}(Q) = \sum_{S\in\{0,1\}^m : M^\intercal S = Q} \widehat{f}(S)$ for all $Q \subseteq [k]$ (where $M^\intercal$ is the transpose of the matrix M).*

Proof. Writing out the Fourier coefficients of $f \circ M$

$$\begin{aligned}
\widehat{f \circ M}(Q) &= \mathop{\mathbb{E}}_{z\in\{0,1\}^k}[(f \circ M)(z)(-1)^{Q\cdot z}] \\
&= \mathop{\mathbb{E}}_{z\in\{0,1\}^k}\Big[\sum_{S\in\{0,1\}^m} \widehat{f}(S)(-1)^{S\cdot(Mz)+Q\cdot z}\Big] \qquad \text{(Fourier expansion of } f\text{)} \\
&= \sum_{S\in\{0,1\}^m} \widehat{f}(S) \mathop{\mathbb{E}}_{z\in\{0,1\}^k}[(-1)^{(M^\intercal S+Q)\cdot z}] \quad \text{(using } \langle S, Mz\rangle = \langle M^\intercal S, z\rangle\text{)} \\
&= \sum_{S:M^\intercal S=Q} \widehat{f}(S). \qquad \text{(using } \mathbb{E}_{z\in\{0,1\}^k}(-1)^{(z_1+z_2)\cdot z} = \delta_{z_1,z_2}\text{)}
\end{aligned}$$

□

[4] Abelian group consists of a set G of elements and an operation $\circ : G^2 \to G$ such that: (i) for every $g_i \neq g_j \neq g_k \in G$, we have $g_i \circ (g_j \circ g_k) = (g_i \circ g_j) \circ g_k$ and $g_i g_j = g_j g_i$, (ii) there exists an identity $e \in G$ such that $e \circ g = g \circ e = g$ for every $g \in G$, (iii) for every $g \in G$, there exists g^{-1} such that $g \circ g^{-1} = g^{-1} \circ g = e$.

7.3.6. Claim. $\max\{(c/\sqrt{t})^t : t \in [1, c^2]\} = e^{c^2/(2e)}$.

Proof. The value of t at which the function $\left(c/\sqrt{t}\right)^t$ is the largest, is obtained by differentiating the function with respect to t,

$$\frac{d}{dt}\left(c/\sqrt{t}\right)^t = (c/\sqrt{t})^t\Big(\ln(c/\sqrt{t}) - 1/2\Big).$$

Equating the derivative to zero we obtain the maxima (the second derivative can be checked to be negative) at $t = c^2/e$. □

7.3.7. Fact. For all $\varepsilon \in [0, 1/2]$ we have $H(\varepsilon) \leq O(\varepsilon \log(1/\varepsilon))$, and (from the Taylor series)

$$1 - H(1/2 + \varepsilon) \leq 2\varepsilon^2/\ln 2 + O(\varepsilon^4).$$

7.3.8. Fact. For every positive integer n, we have that $\binom{n}{k} \leq 2^{nH(k/n)}$ for all $k \leq n$ and $\sum_{i=0}^{m}\binom{n}{i} \leq 2^{nH(m/n)}$ for all $m \leq n/2$.

The following facts are well-known in quantum information theory, which can be found for instance in [KLM06, Theorem A.9.1]

7.3.9. Fact. Let binary random variable $\mathbf{b} \in \{0, 1\}$ be uniformly distributed. Suppose an algorithm is given $|\psi_{\mathbf{b}}\rangle$ (for unknown b) and is required to guess whether $\mathbf{b} = 0$ or $\mathbf{b} = 1$. It will guess correctly with probability at most $\frac{1}{2} + \frac{1}{2}\sqrt{1 - |\langle\psi_0|\psi_1\rangle|^2}$.

Note that if we can distinguish $|\psi_0\rangle$ and $|\psi_1\rangle$ with probability $\geq 1 - \delta$, then $|\langle\psi_0|\psi_1\rangle| \leq 2\sqrt{\delta(1-\delta)}$.

7.3.10. Fact. (Subadditivity of quantum entropy): For an arbitrary bipartite state ρ_{AB} on the Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$, it holds that $S(\rho_{AB}) \leq S(\rho_A) + S(\rho_B)$.

7.4 Information-theoretic lower bounds

Upper bounds on sample complexity carry over from classical to quantum PAC learning, because a quantum example becomes a classical example if we just measure it. Our main goal is to show that the *lower* bounds also carry over. All our lower bounds will involve two terms, one that is independent of $\mathcal{C}$ and one that is dependent on the VC dimension of $\mathcal{C}$. In Section 7.4.1 we prove the VC-independent part of the lower bounds for the *quantum* setting (which also is a lower bound for the classical setting), in Section 7.4.2 we present an information-theoretic lower bound on sample complexity for PAC learning and agnostic learning which yields optimal VC-dependent bounds in the classical case. Using similar ideas, in Section 7.4.4 we obtain near-optimal bounds in the quantum case.

7.4.1 VC-independent part of lower bounds

7.4.1. Lemma ([AS05]). *Let $\mathcal{C}$ be a non-trivial concept class.*[5] *For every $\delta \in (0, 1/2)$, $\varepsilon \in (0, 1/4)$, a (ε, δ)-PAC quantum learner for $\mathcal{C}$ has sample complexity $\Omega(\frac{1}{\varepsilon} \log \frac{1}{\delta})$.*

Proof. Since $\mathcal{C}$ is non-trivial, we may assume there are two concepts $c_1, c_2 \in \mathcal{C}$ defined on two inputs $\{x_1, x_2\}$ as follows $c_1(x_1) = c_2(x_1) = 0$ and $c_1(x_2) = 0$, $c_2(x_2) = 1$. Consider the distribution $D(x_1) = 1-\varepsilon$ and $D(x_2) = \varepsilon$. For $i \in \{1, 2\}$, the state of the algorithm after T queries to $\mathrm{QPEX}(c_i, D)$ is

$$|\psi_i\rangle = \Big(\sqrt{1-\varepsilon}|x_1, 0\rangle + \sqrt{\varepsilon}|x_2, c_i(x_2)\rangle\Big)^{\otimes T}.$$

It follows that $\langle\psi_1|\psi_2\rangle = (1-\varepsilon)^T$. Since the success probability of an (ε, δ)-PAC quantum learner is $\geq 1 - \delta$, Fact 7.3.9 implies $\langle\psi_1|\psi_2\rangle \leq 2\sqrt{\delta(1-\delta)}$. Hence $T = \Omega(\frac{1}{\varepsilon} \log \frac{1}{\delta})$. □

7.4.2. Lemma. *Let $\mathcal{C}$ be a non-trivial concept class. For every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/4)$, a (ε, δ)-agnostic quantum learner for $\mathcal{C}$ has sample complexity $\Omega(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$.*

Proof. Since $\mathcal{C}$ is non-trivial, we may assume there are two concepts $c_1, c_2 \in \mathcal{C}$ and there exists an input $x \in \{0, 1\}^n$ such that $c_1(x) \neq c_2(x)$. Consider the two distributions D_- and D_+ defined as follows: $D_\pm(x, c_1(x)) = (1 \pm \varepsilon)/2$ and $D_\pm(x, c_2(x)) = (1 \mp \varepsilon)/2$. Let $|\psi_\pm\rangle$ be the state after T queries to $\mathrm{QAEX}(D_\pm)$, i.e.,

$$|\psi_\pm\rangle = \Big(\sqrt{(1 \pm \varepsilon)/2}|x, c_1(x)\rangle + \sqrt{(1 \mp \varepsilon)/2}|x, c_2(x)\rangle\Big)^{\otimes T}.$$

It follows that $\langle\psi_+|\psi_-\rangle = (1 - \varepsilon^2)^{T/2}$. Since the success probability of an (ε, δ)-agnostic quantum learner is $\geq 1 - \delta$, Fact 7.3.9 implies $\langle\psi_+|\psi_-\rangle \leq 2\sqrt{\delta(1-\delta)}$. Hence $T = \Omega(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$ □

7.4.2 Optimal lower bound on classical PAC sample complexity

7.4.3. Theorem. *Let $\mathcal{C}$ be a concept class with $\mathrm{VC\text{-}dim}(\mathcal{C}) = d + 1$. Then for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/4)$, every (ε, δ)-PAC learner for $\mathcal{C}$ has sample complexity $\Omega\Big(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\Big)$.*

[5] We refer to a concept class $\mathcal{C}$ as being *trivial* if either $\mathcal{C}$ contains only one concept, or $\mathcal{C}$ contains two concepts c_0, c_1 with $c_0(x) = 1 - c_1(x)$ for every $x \in \{0, 1\}^n$.

Proof. Consider an (ε, δ)-PAC learner for $\mathcal{C}$ that uses T examples. The d-independent part of the lower bound, $T = \Omega(\log(1/\delta)/\varepsilon)$, even holds for quantum examples and was proven in Lemma 7.4.1. Hence it remains to prove $T = \Omega(d/\varepsilon)$. It suffices to show this for a specific distribution D, defined as follows. Let $\mathcal{S} = \{s_0, s_1, \ldots, s_d\} \subseteq \{0,1\}^n$ be some $(d+1)$-element set shattered by $\mathcal{C}$. Define

$$D(s_0) = 1 - 4\varepsilon \text{ and } D(s_i) = 4\varepsilon/d \quad \text{for all } i \in [d].$$

Because $\mathcal{S}$ is shattered by $\mathcal{C}$, for each string $a \in \{0,1\}^d$, there exists a concept $c_a \in \mathcal{C}$ such that $c_a(s_0) = 0$ and $c_a(s_i) = a_i$ for all $i \in [d]$. We define two correlated random variables $\mathbf{A}$ and $\mathbf{B}$ corresponding to the concept and to the examples, respectively. Let $\mathbf{A}$ be a random variable that is uniformly distributed over $\{0,1\}^d$; if $\mathbf{A} = a$, let $\mathbf{B} = \mathbf{B}_1 \ldots \mathbf{B}_T$ be T i.i.d. examples from c_a according to D. We give the following three-step analysis of these random variables:

1. $I(\mathbf{A} : \mathbf{B}) \geq (1-\delta)(1 - H(1/4))d - H(\delta) = \Omega(d)$.
   Proof. Let random variable $h(\mathbf{B}) \in \{0,1\}^d$ be the hypothesis that the learner produces (given the examples in $\mathbf{B}$) restricted to the shattered set $s_1, \ldots, s_d$. Note that the error of the hypothesis $\mathrm{err}_D(h(\mathbf{B}), c_{\mathbf{A}})$ equals $d_H(\mathbf{A}, h(\mathbf{B})) \cdot 4\varepsilon/d$, because each s_i where $\mathbf{A}$ and $h(\mathbf{B})$ differ contributes $D(s_i) = 4\varepsilon/d$ to the error. Let $\mathbf{Z}$ be the indicator random variable for the event that the error is $\leq \varepsilon$. If $\mathbf{Z} = 1$, then $d_H(\mathbf{A}, h(\mathbf{B})) \leq d/4$. Since we are analyzing an (ε, δ)-PAC learner, we have $\Pr[\mathbf{Z} = 1] \geq 1 - \delta$, and $H(\mathbf{Z}) \leq H(\delta)$. Given a string $h(\mathbf{B})$ that is $d/4$-close to $\mathbf{A}$, $\mathbf{A}$ ranges over a set of only $\sum_{i=0}^{d/4} \binom{d}{i} \leq 2^{H(1/4)d}$ possible d-bit strings (using Fact 7.3.8), hence $H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z} = 1) \leq H(\mathbf{A} \mid h(\mathbf{B}), \mathbf{Z} = 1) \leq H(1/4)d$. We now lower bound $I(\mathbf{A} : \mathbf{B})$ as follows:

$$\begin{aligned}
I(\mathbf{A} : \mathbf{B}) &= H(\mathbf{A}) - H(\mathbf{A} \mid \mathbf{B}) \\
&\geq H(\mathbf{A}) - H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z}) - H(\mathbf{Z}) \\
&= H(\mathbf{A}) - \Pr[\mathbf{Z} = 1] \cdot H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z} = 1) \\
&\qquad - \Pr[\mathbf{Z} = 0] \cdot H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z} = 0) - H(\mathbf{Z}) \\
&\geq d - (1-\delta)H(1/4)d - \delta d - H(\delta) \\
&= (1-\delta)(1 - H(1/4))d - H(\delta).
\end{aligned}$$

2. $I(\mathbf{A} : \mathbf{B}) \leq T \cdot I(\mathbf{A} : \mathbf{B}_1)$.
   Proof. This inequality is essentially due to Jain and Zhang [JZ09, Lemma 5],

we include the proof for completeness.

$$\begin{aligned} I(\mathbf{A}:\mathbf{B}) = H(\mathbf{B}) - H(\mathbf{B} \mid \mathbf{A}) &= H(\mathbf{B}) - \sum_{i=1}^{T} H(\mathbf{B}_i \mid \mathbf{A}) \\ &\le \sum_{i=1}^{T} H(\mathbf{B}_i) - \sum_{i=1}^{T} H(\mathbf{B}_i \mid \mathbf{A}) \\ &= \sum_{i=1}^{T} I(\mathbf{A}:\mathbf{B}_i), \end{aligned}$$

where the second equality used independence of the $\mathbf{B}_i$'s conditioned on $\mathbf{A}$, and the inequality uses Fact 7.3.10. Since $I(\mathbf{A}:\mathbf{B}_i) = I(\mathbf{A}:\mathbf{B}_1)$ for all i, we get the inequality.

3. $I(\mathbf{A}:\mathbf{B}_1) = 4\varepsilon$.
   Proof. View $\mathbf{B}_1 = (\mathbf{I}, \mathbf{L})$ as consisting of an index $\mathbf{I} \in \{0, 1, \ldots, d\}$ and a corresponding label $\mathbf{L} \in \{0, 1\}$. With probability $1 - 4\varepsilon$, $(\mathbf{I}, \mathbf{L}) = (0, 0)$. For each $i \in [d]$, with probability $4\varepsilon/d$, $(\mathbf{I}, \mathbf{L}) = (i, \mathbf{A}_i)$. Note that $I(\mathbf{A}:\mathbf{I}) = 0$ because $\mathbf{I}$ is independent of $\mathbf{A}$; $I(\mathbf{A}:\mathbf{L} \mid \mathbf{I} = 0) = 0$; and $I(\mathbf{A}:\mathbf{L} \mid \mathbf{I} = i) = I(\mathbf{A}_i:\mathbf{L} \mid \mathbf{I} = i) = H(\mathbf{A}_i \mid \mathbf{I} = i) - H(\mathbf{A}_i \mid \mathbf{L}, \mathbf{I} = i) = 1 - 0 = 1$ for all $i \in [d]$. We have

$$I(\mathbf{A}:\mathbf{B}_1) = I(\mathbf{A}:\mathbf{I}) + I(\mathbf{A}:\mathbf{L} \mid \mathbf{I}) = \sum_{i=1}^{d} \Pr[\mathbf{I} = i] \cdot I(\mathbf{A}:\mathbf{L} \mid \mathbf{I} = i) = 4\varepsilon.$$

Combining these three steps implies $T = \Omega(d/\varepsilon)$. □

7.4.3 Optimal lower bound on classical agnostic sample complexity

7.4.4. Theorem. *Let $\mathcal{C}$ be a concept class with* VC-dim$(\mathcal{C}) = d$. *Then for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/4)$, every (ε, δ)-agnostic learner for $\mathcal{C}$ has sample complexity $\Omega\left(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\right)$.*

Proof. The d-independent part of the lower bound, $T = \Omega(\log(1/\delta)/\varepsilon^2)$, even holds for quantum examples and was proven in Lemma 7.4.2. For the other part, the proof is similar to Theorem 7.4.3, as follows. Assume an (ε, δ)-agnostic learner for $\mathcal{C}$ that uses T examples. We need to prove $T = \Omega(d/\varepsilon^2)$. For shattered set $\mathcal{S} = \{s_1, \ldots, s_d\} \subseteq \{0, 1\}^n$ and $a \in \{0, 1\}^d$, define distribution D_a by

$$D_a(i, \ell) = (1 + (-1)^{a_i + \ell} 4\varepsilon)/2d \quad \text{for all } (i, \ell) \in [d] \times \{0, 1\}.$$

Again let random variable $\mathbf{A} \in \{0,1\}^d$ be uniformly random, corresponding to the values of concept c_a on $\mathcal{S}$, and $\mathbf{B} = \mathbf{B}_1 \dots \mathbf{B}_T$ be T i.i.d. samples from D_a. Note that c_a is the minimal-error concept from $\mathcal{C}$ w.r.t. D_a, and concept $c_{\tilde{a}}$ has additional error $d_H(a,\tilde{a}) \cdot 4\varepsilon/d$. Accordingly, an (ε,δ)-agnostic learner has to produce (from $\mathbf{B}$) an $h(\mathbf{B}) \in \{0,1\}^d$, which, with probability at least $1-\delta$, is $d/4$-close to $\mathbf{A}$. Our three-step analysis is very similar to Theorem 7.4.3; only the third step changes:

1. $I(\mathbf{A}:\mathbf{B}) \geq (1-\delta)(1-H(1/4))d - H(\delta) = \Omega(d)$.
2. $I(\mathbf{A}:\mathbf{B}) \leq T \cdot I(\mathbf{A}:\mathbf{B}_1)$.
3. $I(\mathbf{A}:\mathbf{B}_1) = 1 - H(1/2+2\varepsilon) = O(\varepsilon^2)$.
   Proof. View the D_a-distributed random variable $\mathbf{B}_1 = (\mathbf{I},\mathbf{L})$ as index $\mathbf{I} \in [d]$ and label $\mathbf{L} \in \{0,1\}$. The marginal distribution of $\mathbf{I}$ is uniform; conditioned on $\mathbf{I} = i$, the bit $\mathbf{L}$ equals $\mathbf{A}_i$ with probability $1/2+2\varepsilon$. Hence
   $$\begin{aligned} I(\mathbf{A}:\mathbf{L} \mid \mathbf{I}=i) = I(\mathbf{A}_i:\mathbf{L} \mid \mathbf{I}=i) &= H(\mathbf{A}_i \mid \mathbf{I}=i) - H(\mathbf{A}_i \mid \mathbf{L},\mathbf{I}=i) \\ &= 1 - H(1/2+2\varepsilon). \end{aligned}$$
   Using Fact 7.3.7, we have
   $$\begin{aligned} I(\mathbf{A}:\mathbf{B}_1) = I(\mathbf{A}:\mathbf{I}) + I(\mathbf{A}:\mathbf{L} \mid \mathbf{I}) &= \sum_{i=1}^{d} \Pr[\mathbf{I}=i] \cdot I(\mathbf{A}:\mathbf{L} \mid \mathbf{I}=i) \\ &= 1 - H(1/2+2\varepsilon) = O(\varepsilon^2). \end{aligned}$$

Combining these three steps implies $T = \Omega(d/\varepsilon^2)$. □

In the theorem below, we optimize the constant in the lower bound of the sample complexity in Theorem 7.4.4. In learning theory such lower bounds are often stated slightly differently. In order to compare the lower bounds, we introduce the following. We first define an *ε-average agnostic learner* for a concept class $\mathcal{C}$ as a learner that, given access to T samples from an $\mathrm{AEX}(D)$ oracle (for some unknown distribution D), needs to output a hypothesis $h_{\mathbf{XY}}$ (where $(\mathbf{X},\mathbf{Y}) \sim D^T$) that satisfies

$$\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})\sim D^T}[\mathrm{err}_D(h_{\mathbf{XY}})] - \mathrm{opt}_D(\mathcal{C}) \leq \varepsilon.$$

Lower bounds on the quantity $(\mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim D^T}[\mathrm{err}_D(h_{\mathbf{XY}})] - \mathrm{opt}_D(\mathcal{C}))$ are generally referred to as *minimax lower bounds* in learning theory. For concept class $\mathcal{C}$, Audibert [Aud08, Aud09] showed that there exists a distribution D, such that if the agnostic learner uses T samples from $\mathrm{AEX}(D)$, then

$$\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})\sim D^T}[\mathrm{err}_D(h_{\mathbf{XY}})] - \mathrm{opt}_D(\mathcal{C}) \geq \frac{1}{6}\sqrt{\frac{d}{T}}.$$

Equivalently, this is a lower bound of $T \geq \frac{d}{36\varepsilon^2}$ on the sample complexity of an ε-average agnostic learner. We obtain a slightly weaker lower bound that is essentially $T \geq \frac{d}{62\varepsilon^2}$:

7.4.5. THEOREM. *Let $\mathcal{C}$ be a concept class with* VC-dim$(\mathcal{C}) = d$. *Then for every $\varepsilon \in (0, 1/10]$, there exists a distribution for which every ε-average agnostic learner has sample complexity at least $\frac{d}{\varepsilon^2} \cdot \left(\frac{1}{62} - \frac{\log(2d+2)}{4d}\right)$.*

Proof. The proof is similar to Theorem 7.4.4. Assume an ε-average agnostic learner for $\mathcal{C}$ that uses T samples. For shattered set $\mathcal{S} = \{s_1, \ldots, s_d\} \subseteq \{0,1\}^n$ and $a \in \{0,1\}^d$, define distribution D_a on $[d] \times \{0,1\}$ by $D_a(i,\ell) = (1 + (-1)^{a_i+\ell}\beta\varepsilon)/2d$, for some constant $\beta \geq 2$ which we shall pick later.

Again let random variable $\mathbf{A} \in \{0,1\}^d$ be uniformly random, corresponding to the values of concept c_a on $\mathcal{S}$, and $\mathbf{B} = \mathbf{B}_1 \ldots \mathbf{B}_T$ be T i.i.d. samples from D_a. Note that c_a is the minimal-error concept from $\mathcal{C}$ w.r.t. D_a, and concept $c_{\tilde{a}}$ has additional error $d_H(a,\tilde{a}) \cdot \beta\varepsilon/d$. Accordingly, an ε-average agnostic learner has to produce (from $\mathbf{B}$) an $h(\mathbf{B}) \in \{0,1\}^d$, which satisfies $\mathbb{E}_{\mathbf{A},\mathbf{B}}[d_H(\mathbf{A}, h(\mathbf{B}))] \leq d/\beta$.

Our three-step analysis is very similar to Theorem 7.4.4; only the first step changes and we discuss that below:

1. $I(\mathbf{A} : \mathbf{B}) \geq d(1 - H(1/\beta)) - \log(d+1)$.

   Proof. Define random variable $\mathbf{Z} = d_H(\mathbf{A}, h(\mathbf{B}))$, then $\mathbb{E}[\mathbf{Z}] \leq d/\beta$. Note that given a string $h(\mathbf{B})$ that is ℓ-close to $\mathbf{A}$, $\mathbf{A}$ ranges over a set of only $\binom{d}{\ell} \leq 2^{H(\ell/d)d}$ possible d-bit strings (using Fact 7.3.8), hence $H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z} = \ell) \leq H(\mathbf{A} \mid h(\mathbf{B}), \mathbf{Z} = \ell) \leq H(\ell/d)d$. We now lower bound $I(\mathbf{A} : \mathbf{B})$

$$\begin{aligned}
I(\mathbf{A} : \mathbf{B}) &= H(\mathbf{A}) - H(\mathbf{A} \mid \mathbf{B}) \\
&\geq H(\mathbf{A}) - H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z}) - H(\mathbf{Z}) \\
&= d - \sum_{\ell=0}^{d+1} \Pr[\mathbf{Z} = \ell] \cdot H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z} = \ell) - H(\mathbf{Z}) \\
&\geq d - \mathop{\mathbb{E}}_{\ell \in \{0,\ldots,d\}}[H(\ell/d)d] - \log(d+1) \qquad (\text{since } \mathbf{Z} \in \{0,\ldots,d\}) \\
&\geq d - dH\left(\frac{\mathbb{E}_\ell[\ell]}{d}\right) - \log(d+1) \qquad (\text{using Jensen's inequality}) \\
&\geq d - dH(1/\beta) - \log(d+1), \qquad (\text{using } \mathbb{E}[\mathbf{Z}] \leq d/\beta)
\end{aligned}$$

   where for the third inequality we used the concavity of the binary entropy function to conclude $\mathbb{E}_\ell[H(\ell/d)] \leq H(\mathbb{E}_\ell[\ell]/d)$ (by Jensen's inequality), and for the fourth inequality we used that $\beta \geq 2$.

2. $I(\mathbf{A} : \mathbf{B}) \leq T \cdot I(\mathbf{A} : \mathbf{B}_1)$.

3. $I(\mathbf{A} : \mathbf{B}_1) = 1 - H(1/2 + \beta\varepsilon/2) \leq \beta^2\varepsilon^2/\ln 4 + O(\varepsilon^4)$ (using Fact 7.3.7 in the inequality).

Combining these three steps implies

$$T \geq \frac{d \ln 4}{\varepsilon^2} \cdot \left(\frac{1 - H(1/\beta)}{\beta^2 + O(\varepsilon^2)} - \frac{\log(d+1)}{\beta^2 d + O(d\varepsilon^2)} \right).$$

Using $\varepsilon \leq 1/10$, $\beta = 4$ to optimize this lower bound, we obtain $T \geq \frac{d}{\varepsilon^2} \cdot \left(\frac{1}{62} - \frac{\log(2d+2)}{4d} \right)$. □

7.4.4 Quantum PAC sample complexity lower bound

Here we will "quantize" the above two classical information-theoretic proofs, yielding lower bounds for quantum sample complexity (in both the PAC and the agnostic setting) that are tight up to a logarithmic factor.

7.4.6. Theorem. *Let $\mathcal{C}$ be a concept class with* VC-dim$(\mathcal{C}) = d + 1$. *Then, for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/4)$, every (ε, δ)-PAC quantum learner for $\mathcal{C}$ has sample complexity $\Omega\left(\frac{d}{\varepsilon \log(d/\varepsilon)} + \frac{\log(1/\delta)}{\varepsilon} \right)$.*

Proof. The proof is analogous to Theorem 7.4.3. We use the same distribution D, with the $\mathbf{B}_i$ now being quantum samples

$$|\psi_a\rangle = \sum_{i \in \{0,1,\ldots,d\}} \sqrt{D(s_i)} |i, c_a(s_i)\rangle.$$

The **AB**-system is now in the following classical-quantum state:

$$\frac{1}{2^d} \sum_{a \in \{0,1\}^d} |a\rangle\langle a| \otimes |\psi_a\rangle\langle\psi_a|^{\otimes T}.$$

The first two steps of our argument are identical to Theorem 7.4.3. We only need to re-analyze step 3:

1. $I(\mathbf{A} : \mathbf{B}) \geq (1 - \delta)(1 - H(1/4))d - H(\delta) = \Omega(d)$.
2. $I(\mathbf{A} : \mathbf{B}) \leq T \cdot I(\mathbf{A} : \mathbf{B}_1)$.
3. $I(\mathbf{A} : \mathbf{B}_1) \leq H(4\varepsilon) + 4\varepsilon \log(2d) = O(\varepsilon \log(d/\varepsilon))$.
   Proof. Since **AB** is a classical-quantum state, we have

   $$I(\mathbf{A} : \mathbf{B}_1) = S(\mathbf{A}) + S(\mathbf{B}_1) - S(\mathbf{A}\mathbf{B}_1) = S(\mathbf{B}_1),$$

   where the first equality follows from definition and the second equality uses $S(\mathbf{A}) = d$ since $\mathbf{A}$ is uniformly distributed in $\{0,1\}^d$, and $S(\mathbf{A}\mathbf{B}_1) = d$

since the matrix $\sigma = \frac{1}{2^d}\sum_{a\in\{0,1\}^d} |a\rangle\langle a| \otimes |\psi_a\rangle\langle\psi_a|$ is block diagonal with 2^d rank-1 blocks on the diagonal. It thus suffices to bound the entropy of the singular values of the reduced state of $\mathbf{B}_1$, which is

$$\rho = \frac{1}{2^d}\sum_{a\in\{0,1\}^d} |\psi_a\rangle\langle\psi_a|.$$

Let $\sigma_0 \geq \sigma_1 \geq \cdots \geq \sigma_{2d} \geq 0$ be its singular values. Since ρ is a density matrix, these form a probability distribution. Note that the upper-left entry of the matrix $|\psi_a\rangle\langle\psi_a|$ is $D(s_0) = 1 - 4\varepsilon$, hence so is the upper-left entry of ρ. This implies $\sigma_0 \geq 1 - 4\varepsilon$. Consider sampling a number $\mathbf{N} \in \{0, 1, \ldots, 2d\}$ according to the σ-distribution. Let $\mathbf{Z}$ be the indicator random variable for the event $\mathbf{N} \neq 0$, which has probability $1 - \sigma_0 \leq 4\varepsilon$. Note that $H(\mathbf{N} \mid \mathbf{Z} = 0) = 0$, because $\mathbf{Z} = 0$ implies $\mathbf{N} = 0$. Also, $H(\mathbf{N} \mid \mathbf{Z} = 1) \leq \log(2d)$, because if $\mathbf{Z} = 1$ then $\mathbf{N}$ ranges over $2d$ elements. We now have

$$\begin{aligned} S(\rho) &= H(\mathbf{N}) = H(\mathbf{N}, \mathbf{Z}) = H(\mathbf{Z}) + H(\mathbf{N} \mid \mathbf{Z}) \\ &= H(\mathbf{Z}) + \Pr[\mathbf{Z} = 0] \cdot H(\mathbf{N} \mid \mathbf{Z} = 0) + \Pr[\mathbf{Z} = 1] \cdot H(\mathbf{N} \mid \mathbf{Z} = 1) \\ &\leq H(4\varepsilon) + 4\varepsilon \log(2d) \\ &= O(\varepsilon \log(d/\varepsilon)). \qquad \text{(using Fact 7.3.7)} \end{aligned}$$

Combining these three steps implies $T = \Omega\Big(\frac{d}{\varepsilon \log(d/\varepsilon)}\Big)$. □

7.4.5 Quantum agnostic sample complexity lower bound

7.4.7. Theorem. *Let $\mathcal{C}$ be a concept class with* VC-dim$(\mathcal{C}) = d$. *Then for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/4)$, every (ε, δ)-agnostic quantum learner for $\mathcal{C}$ has sample complexity $\Omega\Big(\frac{d}{\varepsilon^2 \log(d/\varepsilon)} + \frac{\log(1/\delta)}{\varepsilon^2}\Big)$.*

Proof. The proof is analogous to Theorem 7.4.4, with the $\mathbf{B}_i$ now being quantum samples for D_a, $|\psi_a\rangle = \sum_{i\in[d],\ell\in\{0,1\}} \sqrt{D_a(i,\ell)}|i,\ell\rangle$. Again we only need to re-analyze step 3:

1. $I(\mathbf{A} : \mathbf{B}) \geq (1 - \delta)(1 - H(1/4))d - H(\delta) = \Omega(d)$.
2. $I(\mathbf{A} : \mathbf{B}) \leq T \cdot I(\mathbf{A} : \mathbf{B}_1)$.
3. $I(\mathbf{A} : \mathbf{B}_1) = O(\varepsilon^2 \log(d/\varepsilon))$.
   Proof of step 3. As in step 3 of the proof of Theorem 7.4.6, it suffices to upper bound the entropy of

$$\rho = \frac{1}{2^d}\sum_{a\in\{0,1\}^d} |\psi_a\rangle\langle\psi_a|.$$

We now lower bound the largest singular value of ρ. Consider $|\psi\rangle = \frac{1}{\sqrt{2d}} \sum_{i\in[d],\ell\in\{0,1\}} |i,\ell\rangle$. Then,

$$\langle\psi|\psi_a\rangle = \frac{1}{d}\sum_{i\in[d]} \frac{1}{2}\Big(\sqrt{1+4\varepsilon}+\sqrt{1-4\varepsilon}\Big) = \frac{1}{2}\Big(\sqrt{1+4\varepsilon}+\sqrt{1-4\varepsilon}\Big)$$
$$\geq 1-2\varepsilon^2-O(\varepsilon^4),$$

where the last inequality used the Taylor series expansion of $\sqrt{1+x}$. This implies that the largest singular value of ρ is at least

$$\langle\psi|\rho|\psi\rangle = \frac{1}{2^d}\sum_{a\in\{0,1\}^d} |\langle\psi|\psi_a\rangle|^2 \geq 1-4\varepsilon^2-O(\varepsilon^4).$$

We can now finish as in step 3 of the proof of Theorem 7.4.6:

$$I(\mathbf{A}:\mathbf{B}_1) \leq S(\rho) \leq H(4\varepsilon^2)+4\varepsilon^2\log(2d) = O(\varepsilon^2\log(d/\varepsilon)),$$

using Fact 7.3.7 in the equality.

Combining these three steps implies $T = \Omega\Big(\frac{d}{\varepsilon^2\log(d/\varepsilon)}\Big)$. □

7.5 A lower bound by analysis of state identification

In this section we present a tight lower bound on quantum sample complexity for both the PAC and the agnostic learning settings, using ideas from Fourier analysis to analyze the performance of the Pretty Good Measurement. The core of both lower bounds is a technical theorem which we prove first.

7.5.1 A technical theorem.

7.5.1. Theorem. *For $m \geq 10$, let $f : \{0,1\}^m \to \mathbb{R}$ be defined as $f(z) = (1-\beta\frac{|z|}{m})^T$ for some $\beta \in (0,1]$ and $T \in [1, m/(e^3\beta)]$. For $k \leq m$, let $M \in \mathbb{F}_2^{m\times k}$ be a matrix with rank k. Suppose $A \in \mathbb{R}^{2^k\times 2^k}$ is defined as $A(x,y) = (f\circ M)(x+y)$ for $x,y \in \{0,1\}^k$, then*

$$\sqrt{A}(x,x) \leq \frac{2\sqrt{e}}{2^{k/2}}\Big(1-\frac{\beta}{2}\Big)^{T/2} e^{11T^2\beta^2/m+\sqrt{Tm\beta}} \qquad \text{for all } x\in\{0,1\}^k.$$

Proof. The structure of the proof is to first diagonalize A, relating its eigenvalues to the Fourier coefficients of f. This allows to calculate the diagonal entries

of $\sqrt{A}$ exactly in terms of those Fourier coefficients. We then upper bound those Fourier coefficients using a combinatorial argument.

We first observe the well-known relation between the eigenvalues of a matrix P defined as $P(x,y) = g(x+y)$ for $x, y \in \{0,1\}^k$ (where the addition in $x+y$ is defined over $\mathbb{F}_2$), and the Fourier coefficients of g.

7.5.2. Claim. *Suppose $g : \{0,1\}^k \to \mathbb{R}$ and $P \in \mathbb{R}^{2^k \times 2^k}$ is defined as $P(x,y) = g(x+y)$, then the eigenvalues of P are $\{2^k \widehat{g}(Q) : Q \in \{0,1\}^k\}$.*

Proof. Let $H \in \mathbb{R}^{2^k \times 2^k}$ be the matrix defined as $H(x,y) = (-1)^{x \cdot y}$ for $x, y \in \{0,1\}^k$. It is easy to see that $H^{-1}(x,y) = (-1)^{x \cdot y}/2^k$. We now show that H diagonalizes P:

$$\begin{aligned}
(HPH^{-1})(x,y) &= \frac{1}{2^k} \sum_{z_1, z_2 \in \{0,1\}^k} (-1)^{z_1 \cdot x + z_2 \cdot y} g(z_1 + z_2) \\
&= \frac{1}{2^k} \sum_{z_1, z_2, Q \in \{0,1\}^k} (-1)^{z_1 \cdot x + z_2 \cdot y} \widehat{g}(Q) (-1)^{Q \cdot (z_1 + z_2)} \\
&= \frac{1}{2^k} \sum_{Q \in \{0,1\}^k} \widehat{g}(Q) \sum_{z_1 \in \{0,1\}^k} (-1)^{(x+Q) \cdot z_1} \sum_{z_2 \in \{0,1\}^k} (-1)^{(y+Q) \cdot z_2} \\
&= 2^k \widehat{g}(x) \delta_{x,y}
\end{aligned}$$

where the second equality used the Fourier expansion of g and the last equality used $\sum_{z \in \{0,1\}^k} [(-1)^{(a+b) \cdot z}] = 2^k \delta_{a,b}$.

The eigenvalues of P are the diagonal entries, $\{2^k \widehat{g}(Q) : Q \in \{0,1\}^k\}$. □

We now relate the diagonal entries of $\sqrt{A}$ to the Fourier coefficients of f:

7.5.3. Claim. *For all $x \in \{0,1\}^k$, we have*

$$\sqrt{A}(x,x) = \frac{1}{2^{k/2}} \sum_{Q \in \{0,1\}^k} \sqrt{\sum_{S \in \{0,1\}^m : M^\intercal S = Q} \widehat{f}(S)}.$$

Proof. Since $A(x,y) = (f \circ M)(x+y)$, by Claim 7.5.2 it follows that H (as defined in the proof of Claim 7.5.2) diagonalizes A and the eigenvalues of A are $\{2^k \widehat{f \circ M}(Q) : Q \in \{0,1\}^k\}$. Hence, we have

$$\sqrt{A} = H^{-1} \cdot \operatorname{diag}\left(\left\{ \sqrt{2^k \widehat{f \circ M}(Q)} : Q \in \{0,1\}^k \right\} \right) \cdot H,$$

and the diagonal entries of $\sqrt{A}$ are

$$\sqrt{A}(x,x) = \frac{1}{2^{k/2}} \sum_{Q \in \{0,1\}^k} \sqrt{\widehat{f \circ M}(Q)} = \frac{1}{2^{k/2}} \sum_{Q \in \{0,1\}^k} \sqrt{\sum_{S \in \{0,1\}^m : M^\intercal S = Q} \widehat{f}(S)},$$

where the second equality used Claim 7.3.5. □

In the following lemma, we give an upper bound on the Fourier coefficients of f, which in turn (from the claim above) gives an upper bound on the diagonal entries of $\sqrt{A}$.

7.5.4. Lemma. *For $\beta \in (0,1]$, the Fourier coefficients of $f : \{0,1\}^m \to \mathbb{R}$ defined as $f(z) = (1 - \beta\frac{|z|}{m})^T$, satisfy*

$$0 \leq \widehat{f}(S) \leq 4e\Big(1 - \frac{\beta}{2}\Big)^T \Big(\frac{T\beta}{m}\Big)^q e^{22T^2\beta^2/m}, \quad \textit{for all } S \textit{ such that } |S| = q.$$

Proof. In order to see why the Fourier coefficients of f are non-negative, we first define the set $U = \{u_x^{\otimes T}\}_{x\in\{0,1\}^m}$ where $u_x = \sqrt{1-\beta}|0,0\rangle + \sqrt{\beta/m}\sum_{i\in[m]}|i, x_i\rangle$. Let V be the $2^m \times 2^m$ Gram matrix for the set U. For $x, y \in \{0,1\}^m$, we have

$$\begin{aligned} V(x,y) = (u_x^* u_y)^T &= \Big(1 - \beta + \frac{\beta}{m}\sum_{i=1}^m \langle x_i|y_i\rangle\Big)^T \\ &= \Big(1 - \beta + \frac{\beta}{m}(m - |x+y|)\Big)^T \\ &= \Big(1 - \beta\frac{|x+y|}{m}\Big)^T = f(x+y). \end{aligned}$$

By Claim 7.5.2, the eigenvalues of the Gram matrix V are $\{2^m\widehat{f}(S) : S \in \{0,1\}^m\}$. Since the Gram matrix is psd, its eigenvalues are non-negative, which implies that $\widehat{f}(S) \geq 0$ for all $S \in \{0,1\}^m$.

We now prove the upper bound in the lemma. By definition,

$$\begin{aligned} \widehat{f}(S) &= \mathop{\mathbb{E}}_{z\in\{0,1\}^m}\Big[\Big(1 - \beta\frac{|z|}{m}\Big)^T(-1)^{S\cdot z}\Big] \\ &= \mathop{\mathbb{E}}_{z\in\{0,1\}^m}\Big[\Big(1 - \frac{\beta}{2} + \frac{\beta}{2m}\sum_{i=1}^m(-1)^{z_i}\Big)^T(-1)^{S\cdot z}\Big] \\ &= \sum_{\ell=0}^T \binom{T}{\ell}\Big(1-\frac{\beta}{2}\Big)^{T-\ell}\Big(\frac{\beta}{2m}\Big)^\ell \mathop{\mathbb{E}}_{z\in\{0,1\}^m}\Big[\sum_{i_1,\ldots,i_\ell=1}^m (-1)^{z\cdot(e_{i_1}+\cdots+e_{i_\ell}+S)}\Big] \\ &= \sum_{\ell=0}^T \binom{T}{\ell}\Big(1-\frac{\beta}{2}\Big)^{T-\ell}\Big(\frac{\beta}{2m}\Big)^\ell \sum_{i_1,\ldots,i_\ell=1}^m 1_{[e_{i_1}+\cdots+e_{i_\ell}=S]} \end{aligned}$$

where the second equality used $|z| = \sum_{i\in[m]}(1-(-1)^{z_i})/2$ and the last equality used $\mathbb{E}_{z\in\{0,1\}^m}[(-1)^{(z_1+z_2)\cdot z}] = \delta_{z_1,z_2}$. We will use the following claim to upper bound the combinatorial sum in the quantity above.

7.5.5. Claim. *Fix $S \in \{0,1\}^m$ with Hamming weight $|S| = q$. For every $\ell \in \{q, \dots, T\}$, we have*

$$\sum_{i_1,\dots,i_\ell=1}^{m} 1_{[e_{i_1}+\cdots+e_{i_\ell}=S]} \le \begin{cases} \ell! \cdot m^{(\ell-q)/2} \Big/ \Big(2^{(\ell-q)/2}((\ell-q)/2)!\Big) & \text{if } (\ell-q) \text{ is even} \\ 0 & \text{otherwise} \end{cases}$$

Proof. Since $|S| = q$, we can write $S = e_{r_1}+\cdots+e_{r_q}$ for distinct $r_1, \dots, r_q \in [m]$. There are $\binom{\ell}{q}$ ways to pick q indices in $(i_1, \dots, i_\ell)$ (without loss of generality, let them be $i_1, \dots, i_q$) and there are $q!$ factorial ways to assign $(r_1, \dots, r_q)$ to $(i_1, \dots, i_q)$. It remains to count the number of ways that we can assign values to the remaining indices $i_{q+1}, \dots, i_\ell$ such that $e_{i_{q+1}} + \cdots + e_{i_\ell} = 0$. If $\ell - q$ is odd, then there is no setting of $i_{q+1}, \dots, i_\ell$ that will satisfy $e_{i_{q+1}} + \cdots + e_{i_\ell} = 0$, so the left-hand side of the claim is equal to 0. From now on assume $\ell - q$ is even. We upper bound the number of such assignments by partitioning the $\ell - q$ indices into pairs and assigning the same value to both indices in each pair.

We first count the number of ways to partition a set of $\ell - q$ indices into subsets of size 2. This number is exactly $(\ell-q)!\Big(2^{(\ell-q)/2}((\ell-q)/2)!\Big)^{-1}$. Furthermore, there are m possible values that can be assigned to the pair of indices in each of the $(\ell-q)/2$ subsets such that $e_i + e_j = 0$ within each subset. Note that assigning m possible values to each pair of indices in the $(\ell-q)/2$ subsets overcounts, but this rough upper bound is sufficient for our purposes.

Combining the three arguments, we conclude

$$\sum_{i_1,\dots,i_\ell=1}^{d} 1_{[e_{i_1}+\cdots+e_{i_\ell}=S]} \le \frac{\binom{\ell}{q} q! \cdot (\ell-q)! \cdot m^{(\ell-q)/2}}{2^{(\ell-q)/2}((\ell-q)/2)!},$$

which yields the claim. □

Continuing with the evaluation of the Fourier coefficient and using the claim above, we have

$$\begin{aligned} \widehat{f}(S) &= \sum_{\ell=0}^{T} \binom{T}{\ell} \Big(1-\frac{\beta}{2}\Big)^{T-\ell} \Big(\frac{\beta}{2m}\Big)^{\ell} \sum_{i_1,\dots,i_\ell=1}^{m} 1_{[e_{i_1}+\cdots+e_{i_\ell}=S]} \\ &\le \sum_{\ell=q}^{T} \binom{T}{\ell} \Big(1-\frac{\beta}{2}\Big)^{T-\ell} \Big(\frac{\beta}{2m}\Big)^{\ell} \ell! \cdot m^{(\ell-q)/2} \Big/ \Big(2^{(\ell-q)/2}\Big(\frac{\ell-q}{2}\Big)!\Big) \\ &= \Big(1-\frac{\beta}{2}\Big)^{T} \Big(\frac{2}{m}\Big)^{q/2} \sum_{\ell=q}^{T} \binom{T}{\ell} \ell! \Big(\frac{\beta}{m(2-\beta)}\Big)^{\ell} \Big(\frac{m}{2}\Big)^{\ell/2} \Big/ \Big(\frac{\ell-q}{2}\Big)!, \end{aligned}$$

where we used Claim 7.5.5 in the inequality. We now use some binomial identities

to upper bound this further

$$\begin{aligned}
\widehat{f}(S) &\le \Big(1-\frac{\beta}{2}\Big)^T \Big(\frac{2}{m}\Big)^{q/2} \sum_{\ell=q}^{T} \Big(T\cdot\frac{\beta}{m}\cdot\sqrt{\frac{m}{2}}\Big)^{\ell} \Big/ \Big(\frac{\ell-q}{2}\Big)! \\
&= \Big(1-\frac{\beta}{2}\Big)^T \Big(\frac{T\beta}{m}\Big)^{q} \sum_{r=0}^{T-q} \Big(\frac{T\beta}{\sqrt{2m}}\Big)^{r} \frac{1}{(r/2)!} && \text{(substituting } r \leftarrow (\ell-q)\text{)} \\
&\le \Big(1-\frac{\beta}{2}\Big)^T \Big(\frac{T\beta}{m}\Big)^{q} \sum_{r=0}^{T-q} \Big(\frac{T\beta}{\sqrt{2m}}\Big)^{r} \frac{e^{r/2}}{(r/2)^{r/2}} && \text{(using } n! \ge (n/e)^n\text{)} \\
&= \Big(1-\frac{\beta}{2}\Big)^T \Big(\frac{T\beta}{m}\Big)^{q} \sum_{r=0}^{T-q} \Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r} \\
&\le \Big(1-\frac{\beta}{2}\Big)^T \Big(\frac{T\beta}{m}\Big)^{q} \sum_{r=0}^{T} \Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r} && \text{(since the summands are } \ge 0\text{)} \\
&= \Big(1-\frac{\beta}{2}\Big)^T \Big(\frac{T\beta}{m}\Big)^{q} \Big(\sum_{r=0}^{\lceil e^3T^2\beta^2/m\rceil} \Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r} + \sum_{r=\lceil e^3T^2\beta^2/m\rceil+1}^{T} \Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r} \Big),
\end{aligned}$$

where the first inequality also used $\beta < 1$ and $\binom{T}{\ell}\ell! \le T^\ell$. Note that by the assumptions of the theorem, we have $T^2e^3\beta^2/m \le T\beta \le T$, which allowed us to split the summation into two pieces in the last equality. At this point, we upper bound both pieces in the last equation separately. For the first piece, using Claim 7.3.6 it follows that $\left(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\right)^r$ is maximized at $r = \lceil T^2\beta^2/m\rceil$. Using this we get

$$\sum_{r=0}^{\lceil e^3T^2\beta^2/m\rceil} \Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r} \le \Big(2+\frac{e^3T^2\beta^2}{m}\Big) e^{\lceil T^2\beta^2/m\rceil/2} \le 2e^{22T^2\beta^2/m+1}, \tag{7.6}$$

where in first inequality we upper bound every term using Claim 7.3.6 and the second inequality uses $2 + x \le 2e^x$ for $x \ge 0$ and $e^3 + 1/2 \le 22$. For the second piece, we use

$$\begin{aligned}
\sum_{r=\lceil e^3T^2\beta^2/m\rceil+1}^{T} \Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r} &\le \sum_{r=\lceil e^3T^2\beta^2/m\rceil+1}^{T} \Big(\frac{1}{e}\Big)^{r} \\
&\le \sum_{r=1}^{T} \Big(\frac{1}{e}\Big)^{r} = \frac{1-e^{-T}}{e-1} \le 2/3.
\end{aligned} \tag{7.7}$$

So we finally get

$$\begin{aligned}
\widehat{f}(S) &\le \Big(1-\frac{\beta}{2}\Big)^T \Big(\frac{T\beta}{m}\Big)^{q} \Big(2e^{22T^2\beta^2/m+1} + 2/3\Big) && \text{(using Eq. (7.6), (7.7))} \\
&\le 4e\Big(1-\frac{\beta}{2}\Big)^T \Big(\frac{T\beta}{m}\Big)^{q} e^{22T^2\beta^2/m} && \text{(since } 22T^2\beta^2/m > 0\text{)}
\end{aligned}$$

□

The theorem follows by putting together Claim 7.5.3 and Lemma 7.5.4:

$$\begin{aligned}
\sqrt{A}(x,x) &= \frac{1}{2^{k/2}} \sum_{Q\in\{0,1\}^k} \sqrt{\sum_{S\in\{0,1\}^m : M^\intercal S = Q} \widehat{f}(S)} && \text{(using Claim 7.5.3)} \\
&\leq \frac{1}{2^{k/2}} \sum_{Q\in\{0,1\}^k} \sum_{S\in\{0,1\}^m : M^\intercal S = Q} \sqrt{\widehat{f}(S)} \\
&= \frac{1}{2^{k/2}} \sum_{S\in\{0,1\}^m} \sqrt{\widehat{f}(S)} \\
&= \frac{1}{2^{k/2}} \sum_{q=0}^{m} \sum_{S\in\{0,1\}^m : |S|=q} \sqrt{\widehat{f}(S)} \\
&\leq \frac{2\sqrt{e}}{2^{k/2}} \left(1-\frac{\beta}{2}\right)^{T/2} e^{11T^2\beta^2/m} \sum_{q=0}^{m} \binom{m}{q} \left(\frac{T\beta}{m}\right)^{q/2} && \text{(using Lemma 7.5.4)} \\
&= \frac{2\sqrt{e}}{2^{k/2}} \left(1-\frac{\beta}{2}\right)^{T/2} e^{11T^2\beta^2/m} \left(1+\sqrt{\frac{T\beta}{m}}\right)^{m} && \text{(using binomial theorem)} \\
&\leq \frac{2\sqrt{e}}{2^{k/2}} \left(1-\frac{\beta}{2}\right)^{T/2} e^{11T^2\beta^2/m+\sqrt{Tm\beta}},
\end{aligned}$$

where the first inequality used the lower bound from Lemma 7.5.4, the second equality used $\cup_Q\{S : M^\intercal S = Q\} = \{0,1\}^m$ since rank(M)=k and the last inequality uses $(1+x)^t \leq e^{xt}$ for $x, t \geq 0$. □

7.5.2 Optimal lower bound for quantum PAC sample complexity

We can now prove our tight lower bound on quantum sample complexity in the PAC model:

7.5.6. Theorem. *Let $\mathcal{C}$ be a concept class with* VC-dim$(\mathcal{C}) = d+1$*, for sufficiently large d. Then for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/20)$, every (ε,δ)-PAC quantum learner for $\mathcal{C}$ has sample complexity $\Omega\left(\frac{d}{\varepsilon} + \frac{1}{\varepsilon}\log\frac{1}{\delta}\right)$.*

Proof. The d-independent part of the lower bound is Lemma 7.4.1. To prove the d-dependent part, define a distribution D on a set $\mathcal{S} = \{s_0, \ldots, s_d\} \subseteq \{0,1\}^n$ that is shattered by $\mathcal{C}$ as follows:

$$D(s_0) = 1 - 20\varepsilon \text{ and } D(s_i) = 20\varepsilon/d \quad \text{for all } i \in [d].$$

Now consider a $[d, k, r]_2$ linear code (for $k \geq d/4$, distance $r \geq d/8$) as shown to exist in Theorem 7.3.4 with the generator matrix $M \in \mathbb{F}_2^{d\times k}$ of rank k. Let $\{Mx : x \in \{0,1\}^k\} \subseteq \{0,1\}^d$ be the set of codewords in this linear code; these satisfy $d_H(Mx, My) \geq d/8$ whenever $x \neq y$. For each $x \in \{0,1\}^k$, let c^x be a concept defined on the shattered set as: $c^x(s_0) = 0$ and $c^x(s_i) = (Mx)_i$ for all $i \in [d]$. The existence of such concepts in $\mathcal{C}$ follows from the fact that $\mathcal{S}$ is shattered by $\mathcal{C}$. From the distance property of the code, we have $\Pr_{s\sim D}[c^x(s) \neq c^y(s)] \geq \frac{20\varepsilon}{d}\frac{d}{8} = 5\varepsilon/2$. This in particular implies that an (ε, δ)-PAC quantum learner that tries to ε-approximate a concept from $\{c^x : x \in \{0,1\}^k\}$ should successfully *identify* that concept with probability at least $1 - \delta$.

We now consider the following state identification problem: for $x \in \{0,1\}^k$, denote

$$|\psi_x\rangle = \sum_{i\in\{0,\ldots,d\}} \sqrt{D(s_i)}|s_i, c^x(s_i)\rangle.$$

Let the (ε, δ)-PAC quantum sample complexity be T. Assume $T \leq d/(20e^3\varepsilon)$, since otherwise $T \geq \Omega(d/\varepsilon)$ and the theorem follows. Suppose the learner has knowledge of the ensemble $\mathcal{E} = \{(2^{-k}, |\psi_x\rangle^{\otimes T}) : x \in \{0,1\}^k\}$, and is given $|\psi_x\rangle^{\otimes T} \in \mathcal{E}$ for a uniformly random x. The learner would like to maximize the average probability of success to identify the given state. For this problem, we prove a lower bound on T using the PGM defined in Section 7.3.2. In particular, we show that using the PGM, if a learner successfully identifies the states in $\mathcal{E}$, then $T = \Omega(d/\varepsilon)$. Since the PGM is the optimal measurement[6] that the learner could have performed, the result follows. The following lemma makes this lower bound rigorous and will conclude the proof of the theorem.

7.5.7. Lemma. *For every $x \in \{0,1\}^k$, let $|\psi_x\rangle = \sum_{i\in\{0,\ldots,d\}} \sqrt{D(s_i)}|s_i, c^x(s_i)\rangle$, and $\mathcal{E} = \{(2^{-k}, |\psi_x\rangle^{\otimes T}) : x \in \{0,1\}^k\}$. Then*[7]

$$P^{PGM}(\mathcal{E}) \leq \frac{4e}{2^{d/4+T\varepsilon}} e^{8800T^2\varepsilon^2/d+4\sqrt{5Td\varepsilon}}.$$

Before we prove the lemma, we first show why it implies the theorem. Since we observed in Section 7.3.2 that $P^{opt}(\mathcal{E}) = P^{PGM}(\mathcal{E})$, a good learner satisfies $P^{PGM}(\mathcal{E}) = \Omega(1)$ (say for $\delta = 1/4$), which in turn implies

$$\Omega(\max\{d, T\varepsilon\}) \leq O(\min\{T^2\varepsilon^2/d, \sqrt{Td\varepsilon}\}).$$

[6] For $x \in \{0,1\}^k$, define unitary $U_{c^x} : |s_i, b\rangle \to |s_i, b + c^x(s_i)\rangle$ for all $i \in \{0,\ldots,d\}$. The ensemble $\mathcal{E}$ is generated by applying $\{U_{c^x}\}_{x\in\{0,1\}^k}$ to $|\varphi\rangle = \sum_{i\in\{0,\ldots,d\}} \sqrt{D(s_i)}|s_i, 0\rangle$. View $c^x = (0, Mx) \in \{0,1\}^{d+1}$ as a concatenated string where Mx is a codeword of the $[d, k, r]_2$ code. Since the 2^k codewords of the $[d, k, r]_2$ code form a linear subspace, $\{U_{c^x}\}_{x\in\{0,1\}^k}$ is an Abelian group. From the discussion in Section 7.3.2, we conclude that the PGM is the optimal measurement for this state identification problem.

[7] We made no attempt to optimize the constants here.

Note that if $T\varepsilon$ maximizes the left-hand side, then $d \leq T\varepsilon$ and hence $T \geq \Omega(d/\varepsilon)$. The remaining cases are $\Omega(d) \leq T^2\varepsilon^2/d$ and $\Omega(d) \leq \sqrt{Td\varepsilon}$. Both these statements give us $T \geq \Omega(d/\varepsilon)$. Hence the theorem follows, and it remains to prove Lemma 7.5.7:

Proof. Let $\mathcal{E}' = \{2^{-k/2}|\psi_x\rangle^{\otimes T} : x \in \{0,1\}^k\}$ and G be the $2^k \times 2^k$ Gram matrix for $\mathcal{E}'$. As we saw in Section 7.3.2, the success probability of identifying the states in the ensemble $\mathcal{E}$ using the PGM is

$$P^{PGM}(\mathcal{E}) = \sum_{x\in\{0,1\}^k} \sqrt{G}(x,x)^2.$$

For all $x, y \in \{0,1\}^k$, the entries of the Gram matrix G can be written as:

$$\begin{aligned} G(x,y) = \frac{1}{2^k}\langle\psi_x|\psi_y\rangle^T &= \frac{1}{2^k}\Big((1-20\varepsilon) + \frac{20\varepsilon}{d}\sum_{i=1}^{d}\langle c^x(s_i)|c^y(s_i)\rangle\Big)^T \\ &= \frac{1}{2^k}\Big((1-20\varepsilon) + \frac{20\varepsilon}{d}(d - d_H(Mx,My))\Big)^T \\ &= \frac{1}{2^k}\Big(1 - \frac{20\varepsilon}{d}d_H(Mx,My)\Big)^T, \end{aligned} \tag{7.8}$$

where Mx, $My \in \{0,1\}^d$ are codewords in the linear code defined earlier. Define $f : \{0,1\}^d \to \mathbb{R}$ as $f(z) = (1 - \frac{20\varepsilon}{d}|z|)^T$, and let $A(x,y) = (f \circ M)(x+y)$ for $x, y \in \{0,1\}^k$. Note that $G = A/2^k$. Since we assumed $T \leq d/(20e^3\varepsilon)$, we can use Theorem 7.5.1 (by choosing $m = d$ and $\beta = 20\varepsilon$) to upper bound the success probability of successfully identifying the states in the ensemble $\mathcal{E}$ using the PGM.

$$\begin{aligned} P^{PGM}(\mathcal{E}) &= \sum_{x\in\{0,1\}^k} \sqrt{G}(x,x)^2 \\ &= \frac{1}{2^k}\sum_{x\in\{0,1\}^k} \sqrt{A}(x,x)^2 && \text{(since } G = A/2^k\text{)} \\ &\leq \frac{4e}{2^k}\Big(1 - \frac{\beta}{2}\Big)^T e^{22T^2\beta^2/d + 2\sqrt{Td\beta}} && \text{(using Theorem 7.5.1)} \\ &= \frac{4e}{2^k}\Big(1 - 10\varepsilon\Big)^T e^{8800T^2\varepsilon^2/d + 4\sqrt{5Td\varepsilon}} && \text{(substituting } \beta = 20\varepsilon\text{)} \\ &\leq \frac{4e}{2^{k+T\varepsilon}} e^{8800T^2\varepsilon^2/d + 4\sqrt{5Td\varepsilon}} && \text{(using } (1-10\varepsilon)^T \leq e^{-10\varepsilon T} \leq 2^{-\varepsilon T}\text{)} \end{aligned}$$

The lemma follows by observing that $k \geq d/4$. □

□

7.5.3 Optimal lower bound for quantum agnostic sample complexity

We now use the same approach to obtain a tight lower bound on quantum sample complexity in the *agnostic* setting.

7.5.8. THEOREM. *Let $\mathcal{C}$ be a concept class with* VC-dim$(\mathcal{C}) = d$*, for sufficiently large d. Then for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/10)$, every (ε, δ)-agnostic quantum learner for $\mathcal{C}$ has sample complexity $\Omega\Big(\frac{d}{\varepsilon^2} + \frac{1}{\varepsilon^2}\log\frac{1}{\delta}\Big)$.*

Proof. The d-independent part of the lower bound is Lemma 7.4.2. For the d-dependent term in the lower bound, consider a $[d, k, r]_2$ linear code (for $k \geq d/4$, distance $r \geq d/8$) as shown to exist in Theorem 7.3.4, with generator matrix $M \in \mathbb{F}_2^{d\times k}$ of rank k. Let $\{Mx : x \in \{0,1\}^k\} \subseteq \{0,1\}^d$ be the set of 2^k codewords in this linear code; these satisfy $d_H(Mx, My) \geq d/8$ whenever $x \neq y$. To each codeword $x \in \{0,1\}^k$ we associate a distribution D_x as follows:

$$D_x(s_i, b) = \frac{1}{d}\Big(\frac{1}{2} + \frac{1}{2}(-1)^{(Mx)_i + b}\alpha\Big), \qquad \text{for } (i, b) \in [d] \times \{0,1\},$$

where $\mathcal{S} = \{s_1, \ldots, s_d\}$ is a set that is shattered by $\mathcal{C}$, and α is a parameter which we shall pick later. Let $c^x \in \mathcal{C}$ be a concept that labels $\mathcal{S}$ according to $Mx \in \{0,1\}^d$. The existence of such $c^x \in \mathcal{C}$ follows from the fact that $\mathcal{S}$ is shattered by $\mathcal{C}$. Note that c^x is the minimal-error concept in $\mathcal{C}$ w.r.t. D_x. A learner that labels $\mathcal{S}$ according to some string $\ell \in \{0,1\}^d$ has additional error $d_H(Mx, \ell)\cdot\alpha/d$ compared to c^x. This in particular implies that an (ε, δ)-agnostic quantum learner has to find (with probability at least $1 - \delta$) an ℓ such that $d_H(Mx, \ell) \leq d\varepsilon/\alpha$. We pick $\alpha = 20\varepsilon$ and we get $d_H(Mx, \ell) \leq d/20$. However, since Mx was a codeword of a $[d, k, r]_2$ code with distance $r \geq d/8$, finding an ℓ satisfying $d_H(Mx, \ell) \leq d/20$ is equivalent to *identifying* Mx, and hence x.

Now consider the following state identification problem: for $x \in \{0,1\}^k$, let

$$|\psi_x\rangle = \sum_{(i,b)\in[d]\times\{0,1\}} \sqrt{D_x(s_i, b)}\,|s_i, b\rangle.$$

Let the (ε, δ)-agnostic quantum sample complexity be T. Furthermore, assume that $T \leq d/(100e^3\varepsilon^2)$, since otherwise $T \geq \Omega(d/\varepsilon^2)$ and the theorem follows. Suppose the learner has knowledge of the ensemble $\mathcal{E} = \{(2^{-k}, |\psi_x\rangle^{\otimes T}) : x \in \{0,1\}^k\}$, and is given $|\psi_x\rangle^{\otimes T} \in \mathcal{E}$ for uniformly random x. The learner would like to maximize the average probability of success to identify the given state. For this problem, we prove a lower bound on T using the PGM defined in Section 7.3.2. In particular, we show that using the PGM, if a learner successfully identifies the

states in $\mathcal{E}$, then $T = \Omega(d/\varepsilon^2)$. Since the PGM is the optimal measurement[8] that the learner could have performed, the result follows. The following lemma makes this lower bound rigorous and will conclude the proof of the theorem.

7.5.9. LEMMA. *For $x \in \{0,1\}^k$, let $|\psi_x\rangle = \sum_{(i,b)\in[d]\times\{0,1\}} \sqrt{D_x(s_i,b)}|s_i,b\rangle$, and $\mathcal{E} = \{(2^{-k}, |\psi_x\rangle^{\otimes T}) : x \in \{0,1\}^k\}$. Then*

$$P^{PGM}(\mathcal{E}) \leq \frac{4e}{e^{(d\ln 2)/4+25T\varepsilon^2}} e^{220000T^2\varepsilon^4/d+20\sqrt{Td\varepsilon^2}}.$$

Before we prove the lemma, we first show why it implies the theorem. Since we observed above that $P^{opt}(\mathcal{E}) = P^{PGM}(\mathcal{E})$, a good learner satisfies $P^{PGM}(\mathcal{E}) = \Omega(1)$ (say for $\delta = 1/4$), which in turn implies

$$\Omega(\max\{d, T\varepsilon^2\}) \leq O(\min\{T^2\varepsilon^4/d, \sqrt{Td\varepsilon^2}\}).$$

Like in the proof of Theorem 7.5.6, this implies a lower bound of $T = \Omega(d/\varepsilon^2)$ and proves the theorem. It remains to prove Lemma 7.5.9:

Proof. Let $\mathcal{E}' = \{2^{-k/2}|\psi_x\rangle^{\otimes T} : x \in \{0,1\}^k\}$ and G be the $2^k \times 2^k$ Gram matrix for the set $\mathcal{E}'$. As we saw in Theorem 7.3.1 in Section 7.3.2, the success probability of identifying the states in the ensemble $\mathcal{E}$ using the PGM is

$$P^{PGM}(\mathcal{E}) = \sum_{x\in\{0,1\}^k} \sqrt{G}(x,x)^2.$$

For all $x, y \in \{0,1\}^k$, the entries of G can be written as:

$$\begin{aligned}
2^k \cdot G(x,y) &= \langle\psi_x|\psi_y\rangle^T \\
&= \Big(\sum_{(i,b)\in[d]\times\{0,1\}} \sqrt{D_x(i,b)D_y(i,b)}\Big)^T \\
&= \Big(\frac{1}{2d}\sum_{(i,b)\in[d]\times\{0,1\}} \sqrt{(1+10\varepsilon(-1)^{(Mx)_i+b})(1+10\varepsilon(-1)^{(My)_i+b})}\Big)^T \\
&= \Big(\frac{1}{2d}\sum_{\substack{(i,b):\\(Mx)_i=(My)_i}} (1+10\varepsilon(-1)^{(Mx)_i+b}) + \frac{1}{2d}\sum_{\substack{(i,b):\\(Mx)_i\neq(My)_i}} \sqrt{1-100\varepsilon^2}\Big)^T \\
&= \Big(\frac{d-d_H(Mx,My)}{d} + \frac{\sqrt{1-100\varepsilon^2}}{d} d_H(Mx,My)\Big)^T \\
&= \Big(1 - \frac{1-\sqrt{1-100\varepsilon^2}}{d} d_H(Mx,My)\Big)^T.
\end{aligned}$$

[8] For $x \in \{0,1\}^k$, define unitary $U_{c^x} = \sum_{i\in[d]} |s_i\rangle\langle s_i| \otimes X^{(Mx)_i}$, where X is the NOT-gate, so $X^{(Mx)_i}|b\rangle = |b + (Mx)_i\rangle$ for $b \in \{0,1\}$. The ensemble $\mathcal{E}$ is generated by applying $\{U_{c^x}\}_{x\in\{0,1\}^k}$ to $|\varphi\rangle = \frac{1}{\sqrt{d}}\sum_{(i,b)\in[d]\times\{0,1\}} \sqrt{\frac{1}{2} + \frac{1}{2}(-1)^b\alpha}|s_i,b\rangle$. Since the 2^k codewords of the $[d,k,r]_2$ code form a linear subspace, $\{U_{c^x}\}_{x\in\{0,1\}^k}$ is an Abelian group. From the discussion in Section 7.3.2, we conclude that the PGM is the optimal measurement for this state identification problem.

where we used $\alpha = 20\varepsilon$ in the third equality.

Let $\beta = 1 - \sqrt{1 - 100\varepsilon^2}$, which is at most 1 for $\varepsilon \leq 1/10$. Define $f : \{0,1\}^d \to \mathbb{R}$ as $f(z) = (1 - \frac{\beta}{d}|z|)^T$, and let $A(x,y) = (f \circ M)(x+y)$ for $x, y \in \{0,1\}^k$. Then $G = A/2^k$. Note that $T \leq d/(100e^3\varepsilon^2) \leq d/(e^3\beta)$ (the first inequality is by assumption and the second inequality follows for $\varepsilon \leq 1/10$ and $\beta \leq 1$). Since we assumed $T \leq d/(100e^3\varepsilon^2)$, we can use Theorem 7.5.1 (by choosing $m = d$ and $\beta = 1 - \sqrt{1 - 100\varepsilon^2}$) to upper bound the success probability of identifying the states in the ensemble $\mathcal{E}$:

$$\begin{aligned}
P^{PGM}(\mathcal{E}) &= \sum_{x \in \{0,1\}^k} \sqrt{G}(x,x)^2 \\
&= \frac{1}{2^k} \sum_{x \in \{0,1\}^k} \sqrt{A}(x,x)^2 && \text{(since } G = A/2^k\text{)} \\
&\leq \frac{4e}{2^k}\left(1 - \frac{\beta}{2}\right)^T e^{22T^2\beta^2/d + 2\sqrt{Td\beta}} && \text{(using Theorem 7.5.1)} \\
&\leq \frac{4e}{2^k}\left(1 - \frac{\beta}{2}\right)^T e^{220000T^2\varepsilon^4/d + 20\sqrt{Td\varepsilon^2}} \\
&\leq \frac{4e}{2^k}\left(1 - 25\varepsilon^2\right)^T e^{220000T^2\varepsilon^4/d + 20\sqrt{Td\varepsilon^2}} && \text{(using } \sqrt{1-100\varepsilon^2} \leq 1 - 50\varepsilon^2\text{)} \\
&\leq \frac{4e}{e^{k\ln 2 + 25T\varepsilon^2}} e^{220000T^2\varepsilon^4/d + 20\sqrt{Td\varepsilon^2}},
\end{aligned}$$

where we used $\beta = 1 - \sqrt{1 - 100\varepsilon^2} \leq 100\varepsilon^2$ in the second inequality and $(1 - x)^t \leq e^{-xt}$ for $x, t \geq 0$ in the last inequality. The lemma follows by observing that $k \geq d/4$. □

□

7.6 Additional results.

In this section we mention two additional results that can also be obtained using our main Theorem 7.5.1.

7.6.1 Lower bound for PAC learning under random classification noise

In the theorem below, we show a lower bound on the quantum PAC sample complexity under the random classification noise model with noise rate η. Recall that in this model, for every $c \in \mathcal{C}$ and distribution D, $\varepsilon, \delta > 0$, given access to

copies of the η-noisy state,

$$\sum_{x\in\{0,1\}^n} \sqrt{(1-\eta)D(x)}|x, c(x)\rangle + \sqrt{\eta D(x)}|x, 1-c(x)\rangle,$$

a (ε,δ)-PAC quantum learner is required to output an hypothesis h such that $\mathrm{err}_D(c,h) \le \varepsilon$ with probability at least $1-\delta$.

7.6.1. THEOREM. *Let $\mathcal{C}$ be a concept class with* VC-dim$(\mathcal{C}) = d+1$*, for sufficiently large d. Then for every $\delta \in (0,1/2)$, $\varepsilon \in (0,1/20)$ and $\eta \in (0,1/2)$, every (ε,δ)-PAC quantum learner for $\mathcal{C}$ in the PAC setting with random classification noise rate η, has sample complexity $\Omega\Big(\frac{d}{(1-2\eta)^2\varepsilon} + \frac{\log(1/\delta)}{(1-2\eta)^2\varepsilon}\Big)$.*

Proof sketch. One can use exactly the same proof technique as in Lemma 7.4.1 and Theorem 7.5.6. We only sketch the inner product calculation in Eq. (7.8) here. Let $\{c^x : x \in \{0,1\}^k\}$ be the 2^k concepts and D be the distribution

$$D(s_0) = 1-20\varepsilon \text{ and } D(s_i) = 20\varepsilon/d \quad \text{for all } i \in [d],$$

as defined in Theorem 7.5.6. The η-noisy quantum examples corresponding to these concepts are

$$|\psi_x\rangle = \sum_{i\in\{0,\ldots,d\}} \big(\sqrt{D(s_i)(1-\eta)}|s_i, c^x(s_i)\rangle + \sqrt{D(s_i)\eta}|s_i, 1-c^x(s_i)\rangle\big).$$

Then,

$$\begin{aligned}
\langle\psi_x|\psi_y\rangle =& (1-\eta)\Big(1-20\varepsilon + \frac{20\varepsilon}{d}\sum_{i\in\{0,\ldots,d\}}\langle c^x(s_i)|c^y(s_i)\rangle\Big)\\
&+\eta\Big(1-20\varepsilon+\frac{20\varepsilon}{d}\sum_{i\in\{0,\ldots,d\}}\langle 1-c^x(s_i)|1-c^y(s_i)\rangle\Big)\\
&+2\sqrt{\eta(1-\eta)}\cdot\frac{20\varepsilon}{d}\sum_{i\in\{0,\ldots,d\}}\langle c^x(s_i)|1-c^y(s_i)\rangle\\
&=1-\frac{20\varepsilon}{d}\Big(1-2\sqrt{\eta(1-\eta)}\Big)d_H(Mx,My),
\end{aligned}$$

where the second equality used $\sum_i\langle c^x(s_i)|c^y(s_i)\rangle = d - d_H(Mx,My)$ as well as $\sum_i\langle c^x(s_i)|1-c^y(s_i)\rangle = d_H(Mx,My)$. Now let $\varepsilon' = \varepsilon(1-2\sqrt{\eta(1-\eta)})$ and carry on with the proof of Theorem 7.5.6. We get a lower bound of

$$T = \Omega\Big(\frac{d}{\varepsilon'}\Big) = \Omega\Big(\frac{d}{\varepsilon(1-2\sqrt{\eta(1-\eta)})}\Big) = \Omega\Big(\frac{d}{\varepsilon(1-2\eta)^2}\Big),$$

where we used $1-2\sqrt{\eta(1-\eta)} \le (1-2\eta)^2$, which holds for $\eta \le 1/2$. □

7.6.2 Distinguishing codeword states

Ashley Montanaro (personal communication) alerted us to the following interesting special case of our PGM-based result.

Consider an $[n,k,d]_2$ linear code $\{Mx : x \in \{0,1\}^k\}$, where $M \in \mathbb{F}_2^{n\times k}$ is the rank-k generator matrix of the code, $k = \Omega(n)$, and distinct codewords have Hamming distance at least d.[9] For every $x \in \{0,1\}^k$, define a *codeword state* $|\psi_x\rangle = \frac{1}{\sqrt{n}} \sum_{i\in[n]} |i,(Mx)_i\rangle$. These states form an example of a *quantum fingerprinting* scheme [BCWW01]: 2^k states whose pairwise inner products are bounded away from 1. How many copies do we need to identify one such fingerprint?

Let $\mathcal{E} = \{(2^{-k}, |\psi_x\rangle) : x \in \{0,1\}^k\}$ be an ensemble of codeword states. Consider the following task: given T copies of an unknown state drawn uniformly from $\mathcal{E}$, we are required to identify the state with probability $\geq 4/5$. From Holevo's theorem one can easily obtain a lower bound of $T = \Omega(k/\log n)$ copies, since the learner should obtain $\Omega(k)$ bits of information (i.e., identify k-bit string x with probability $\geq 4/5$), while each copy of the codeword state gives at most $\log n$ bits of information. In the theorem below, we improve that $\Omega(k/\log n)$ to the optimal $\Omega(k)$ for constant-rate codes.

7.6.2. Theorem. *Let $\mathcal{E} = \{|\psi_x\rangle = \frac{1}{\sqrt{n}} \sum_{i\in[n]} |i,(Mx)_i\rangle : x \in \{0,1\}^k\}$, where $M \in \mathbb{F}_2^{n\times k}$ is the generator matrix of an $[n,k,d]_2$ linear code with $k = \Omega(n)$. Then $\Omega(k)$ copies of an unknown state from $\mathcal{E}$ (drawn uniformly at random) are necessary to be able to identify that state with probability at least $4/5$.*

One can use exactly the proof technique of Theorem 7.5.6 to prove the theorem. Suppose we are given T copies of the unknown codeword state. Assume $T \leq n$, since otherwise $T \geq n \geq \sqrt{kn}$ and the theorem follows. Observe that the Gram matrix G for $\mathcal{E}' = \{2^{-k/2}|\psi_x\rangle^{\otimes T} : x \in \{0,1\}^k\}$ can be written as $G(x,y) = \frac{1}{2^k}\left(1 - \frac{|M(x+y)|}{n}\right)^T$ for $x,y \in \{0,1\}^k$. Using Theorem 7.5.1 (choosing $\beta = 1$ and $m = n$) to upper bound the success probability of successfully identifying the states in the ensemble $\mathcal{E}$ using the PGM, we obtain

$$P^{PGM}(\mathcal{E}) \leq \frac{4e}{2^{k+T}} e^{22T^2/n + 2\sqrt{Tn}}.$$

As in the proof of Theorem 7.5.6, this implies the lower bound of Theorem 7.6.2. We omit the details of the calculation.

[9] Note that throughout this chapter $\mathcal{C}$ was a concept class in $\{0,1\}^n$ and d was the VC dimension of $\mathcal{C}$. The use of n, d in this section has been changed to conform to the convention in coding theory.

7.7 Conclusion and future work

The main result of this chapter is that quantum examples give no significant improvement over the usual random examples in passive, distribution-independent settings. Of course, these negative results do not mean that quantum machine learning is useless. In the previous chapter we already mentioned improvements from quantum examples for learning under the uniform distribution; improvements from using quantum membership queries; and improvements in time complexity based on quantum algorithms like Grover's and HHL. Quantum machine learning is still in its infancy, and we hope for many more positive results.

We end by identifying a number of open questions for future work:

- We gave lower bounds on sample complexity for the rather benign random classification noise. What about other noise models, such a *malicious* noise?
- What is the quantum sample complexity for learning concepts whose range is $[k]$ rather than $\{0,1\}$, for some $k > 2$? Even the *classical* sample complexity is not fully determined yet [SB14, Section 29.2].
- Classically, it is still an open question whether the $\log(1/\varepsilon)$-factor in the upper bound of [BEHW89] for (ε,δ)-*proper* PAC learning is necessary. A weaker result (possibly easier to prove) would be to give a (ε,δ)-*quantum* proper PAC learner without this $\log(1/\varepsilon)$-factor.
- In the introduction we mentioned a few examples of learning under the *uniform* distribution where quantum examples are significantly more powerful than classical examples. Can we find more such examples of quantum improvements in sample complexity in fixed-distribution settings?

Bibliography

[AA05] S. Aaronson and A. Ambainis. Quantum search of spatial regions. *Theory of Computing*, 1(1):47–79, 2005. Earlier version in FOCS'03. quant-ph/0303041. [39]

[AA15] S. Aaronson and A. Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC*, pages 307–316, 2015. arXiv:1411.5729v1. [5,25,55,78]

[AAD+15] J. Adcock, E. Allen, M. Day, S. Frick, J. Hinchliff, M. Johnson, S. Morley-Short, S. Pallister, A. Price, and S. Stanisic. Advances in quantum machine learning, 2015. arXiv:1512.02900. [8,111]

[AAI+16] S. Aaronson, A. Ambainis, J. Iraids, M. Kokainis, and J. Smotrovs. Polynomials, quantum query complexity, and Grothendieck's inequality. In *31st Conference on Computational Complexity, CCC 2016*, pages 25:1–25:19, 2016. arXiv:1511.08682. [5,53,55,56,59,68,78]

[Aar05a] S. Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, 1(1):1–28, 2005. arXiv:quant-ph/0402095. [130]

[Aar05b] S. Aaronson. Ten semi-grand challenges for quantum computing theory. `http://www.scottaaronson.com/writings/qchallenge.html`, 2005. [137]

[Aar07] S. Aaronson. The learnability of quantum states. *Proceedings of the Royal Society of London*, 463(2088), 2007. quant-ph/0608142. [128,132]

[Aar15] S. Aaronson. Quantum machine learning algorithms: Read the fine print. *Nature Physics*, 11(4):291–293, April 2015. [111]

[Aar17] S. Aaronson. Shadow tomography of quantum states, 2017. arXiv:1711.01053v1. [129,130]

[AB00] M. Anthony and P. Bartlett. Function learning from interpolation. *Combinatorics, Probability, and Computing*, 9(3):213–225, 2000. Earlier version in EuroCOLT'95. [129]

[AB09] M. Anthony and P. L. Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 2009. [110,115,144]

[ABB+16] A. Ambainis, K. Balodis, A. Belovs, T. Lee, M. Santha, and J. Smotrovs. Separations in query complexity based on pointer functions. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 800–813, 2016. arXiv:1506.04719. [13,25]

[ABG06] E. Aïmeur, G. Brassard, and S. Gambs. Machine learning in a quantum world. In *Proceedings of Advances in Artificial Intelligence, 19th Conference of the Canadian Society for Computational Studies of Intelligence*, volume 4013, pages 431–442, 2006. [110]

[ABG13] E. Aïmeur, G. Brassard, and S. Gambs. Quantum speed-up for unsupervised learning. *Machine Learning*, 90(2):261–287, 2013. [110]

[ABK16] S. Aaronson, S. Ben-David, and R. Kothari. Separations in query complexity using cheat sheets. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 863–876, 2016. arXiv:1511.01937. [13,55]

[ABP18] S. Arunachalam, J. Briët, and C. Palazuelos. Quantum query algorithms are completely bounded forms. In *Innovations in Theoretical Computer Science (ITCS)*, pages 3:1–3:21, 2018. arXiv:1711.07285. [v,53]

[AC17] S. Aaronson and L. Chen. Complexity-theoretic foundations of quantum supremacy experiments. In *32nd Computational Complexity Conference (CCC)*, pages 22:1–22:67, 2017. arXiv:1612.05903. [138]

[ACLW18] S. Arunachalam, S. Chakraborty, T. Lee, and R. de Wolf. Two new results on quantum exact learning. Manuscript, 2018. [v,120,136]

[ACR+10] A. Ambainis, A. M. Childs, B. W. Reichardt, R. Špalek, and S. Zhang. Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. *SIAM Journal on Computing*, 39(6):2513–2530, 2010. Earlier version in FOCS'07 and arXiv:quant-ph/0703015. [82,84]

[AG98] B. Apolloni and C. Gentile. Sample size lower bounds in PAC learning by algorithmic complexity theory. *Theoretical Computer Science*, 209:141–162, 1998. [144]

[AGGW17] J. van Apeldoorn, A. Gilyén, S. Gribling, and R. de Wolf. Quantum SDP-solvers: Better upper and lower bounds. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 403–414, 2017. arXiv:1705.01843. [83,84,89,105]

[AGJO+15] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, and P. V. Srinivasan. On the robustness of bucket brigade quantum RAM. *New Journal of Physics*, 17(12):123010, 2015. arXiv:1502.03450. Also in TQC 2015. [v]

[AIK+04] A. Ambainis, K. Iwama, A. Kawachi, H. Masuda, R. H. Putra, and S. Yamashita. Quantum identification of Boolean oracles. In *Proceedings of 30th Annual Symposium on Theoretical Aspects of Computer Science (STACS'04)*, pages 105–116, 2004. arXiv:quant-ph/0403056. [121]

[AIK+07] A. Ambainis, K. Iwama, A. Kawachi, R. Raymond, and S. Yamashita. Improved algorithms for quantum identification of Boolean oracles. *Theoretical Computer Science*, 378(1):41–53, 2007. [121]

[AIN+09] A. Ambainis, K. Iwama, M. Nakanishi, H. Nishimura, R. Raymond, S. Tani, and S. Yamashita. Average/worst-case gap of quantum query complexities by on-set size. 2009. arXiv:0908.2468v1. [121]

[AJR15] S. Arunachalam, N. Johnston, and V. Russo. Is absolute separability determined by the partial transpose? *Quantum Information & Computation*, 15(7-8):694–720, 2015. arXiv:1405.5853. [v]

[AK95] D. Angluin and M. Kharitonov. When won't membership queries help? *Journal of Computer and System Sciences*, 50(2):336–355, 1995. Earlier version in STOC'91. [132]

[AL88] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988. [145]

[AM14] A. Ambainis and A. Montanaro. Quantum algorithms for search with wildcards and combinatorial group testing. *Quantum Information & Computation*, 14(5-6):439–453, 2014. arXiv:1210.1148. 146

[Amb00] A. Ambainis. Quantum lower bounds by quantum arguments. In *STOC*, pages 636–643, 2000. quant-ph/0002066. 5,25,28,29,30,119,196,199

[Amb02] A. Ambainis. Quantum lower bounds by quantum arguments. *J. Comput. Syst. Sci.*, 64(4):750–767, 2002. Earlier version in STOC'00. arXiv:quant-ph/0002066. 5,29

[Amb04] A. Ambianis. Quatum search algorithms. *ACM SIGACT News*, 35(2):22–35, 2004. arXiv:quant-ph/0504012. 31

[Amb06] A. Ambainis. Polynomial degree vs. quantum query complexity. *J. Comput. System Sci.*, 72(2):220–238, 2006. Earlier version in FOCS'03. quant-ph/0305028. 5,55

[Amb07] A. Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007. Earlier version in FOCS'04. arXiv:quant-ph/0311001. 25,82

[AMR17] S. Arunachalam, A. Molina, and V. Russo. Quantum hedging in two-round prover-verifier interactions. In *12th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC)*, 2017. arXiv:1310.7954. v

[AN06] N. Alon and A. Naor. Approximating the cut-norm via Grothendieck's inequality. *SIAM Journal of Computing*, 35(4):787–803, 2006. Earlier version in STOC'04. 59,75

[Ang87] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1987. 114,138

[ANTV02] A. Ambainis, A. Nayak, A. Ta-Shma, and U. V. Vazirani. Dense quantum coding and quantum finite automata. *Journal of the ACM*, 49(4):496–511, 2002. Earlier version in STOC'99. 129

[APVZ14] A. Andoni, R. Panigrahy, G. Valiant, and L. Zhang. Learning sparse polynomial functions. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 500–510, 2014. 136

[Aru14] S. Arunachalam. Quantum speed-ups for Boolean satisfiability and derivative-free optimization. Master's thesis, University of Waterloo, 2014. 6,84

[Arv12] W. Arveson. *An invitation to C^*-algebras*, volume 39 of *Graduate Texts in Mathematics*. Springer, 2012. 61

[AS04] S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004. 5,55

[AS05] A. Atıcı and R. Servedio. Improved bounds on quantum learning algorithms. *Quantum Information Processing*, 4(5):355–386, 2005. quant-ph/0411140. 8,9,110,124,126,138,143,151

[AS09] A. Atıcı and R. Servedio. Quantum algorithms for learning and testing juntas. *Quantum Information Processing*, 6(5):323–348, 2009. arXiv:0707.3479. 110,135,136,143

[Aud08] J. Audibert. Fast learning rates in statistical inference through aggregation, 2008. Research Report 06-20, CertisEcole des Ponts. math/0703854. 154

[Aud09] J. Audibert. Fast learning rates in statistical inference through aggregation. *The Annals of Statistics*, 37(4):1591–1646, 2009. arXiv:0909.1468v1. 144,154

[AW17a] S. Arunachalam and R. de Wolf. Guest column: A survey of quantum learning theory. *SIGACT News*, 48(2):41–67, 2017. arXiv:1701.06806. v,8,109,111

[AW17b] S. Arunachalam and R. de Wolf. Optimal quantum sample complexity of learning algorithms. In *32nd Computational Complexity Conference, CCC 2017*, pages 25:1–25:31, 2017. arXiv:1607.00932. v,110,117,126,127,137,139

[AW17c] S. Arunachalam and R. de Wolf. Optimizing the number of gates in quantum search. *Quantum Information & Computation*, 17(3&4):251–261, 2017. arXiv:1512.07550. v,37

[Azo94] E. M. Azoff. *Neural Network Time Series Forecasting of Financial Markets*. John Wiley & Sons, New York, NY, USA, 1st edition, 1994. 103

[Bal12] P. Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Unsupervised and Transfer Learning - Workshop held at ICML 2011*, pages 37–50, 2012. 103

[BBBV97] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. quant-ph/9701001. 4,34,37,55,102

[BBC+01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Earlier version in FOCS'98. quant-ph/9802049. 5,22,25,26,53,54,55,196,199

[BBLV12] J. Briët, H. Buhrman, T. Lee, and T. Vidick. All Schatten spaces endowed with the Schur product are Q-algebras. *Journal of Functional Analysis*, 262(1):1–9, 2012. 60

[BBLV13] J. Briët, H. Buhrman, T. Lee, and T. Vidick. Multipartite entanglement in XOR games. *Quantum Information & Computation*, 13(3-4):334–360, 2013. arXiv:0911.4007. 60

[BCC+15] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Physical Review Letters*, 114:090502, 2015. 91

[BCD06] D. Bacon, A. Childs, and W. van Dam. Optimal measurements for the dihedral hidden subgroup problem. *Chicago Journal of Theoretical Computer Science*, 2006. Earlier version in FOCS'05. quant-ph/0504083. 146

[BCG+96] N. H. Bshouty, R. Cleve, R. Gavaldà, S. Kannan, and C. Tamon. Oracles and queries that are sufficient for exact learning. *Journal of Computer and System Sciences*, 52(3):421–433, 1996. Earlier version in COLT'94. 118,119

[BCK15] D. W. Berry, A. M. Childs, and R. Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS*, pages 792–809, 2015. arXiv:1501.01715. 84

[BCW98] H. Buhrman, R. Cleve, and A. Wigderson. Quantum vs. classical communication and computation. In *Proceedings of 30th ACM STOC*, pages 63–68, 1998. quant-ph/9802040. 31

[BCWW01] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16), 2001. quant-ph/0102001. 170

[BDH+05] H. Buhrman, C. Dürr, M. Heiligman, P. Høyer, F. Magniez, M. Santha, and R. de Wolf. Quantum algorithms for element distinctness. *SIAM Journal on Computing*, 34(6):1324–1330, 2005. Earlier version in CCC'01. quant-ph/0007016. 31

[BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989. 125,126,141,149,171

[Bel12] A. Belovs. Span programs for functions with constant-sized 1-certificates. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012*, pages 77–84, 2012. arXiv:1105.4024. 25,81

[Bel13] A. Belovs. Quantum algorithms for learning symmetric juntas via adversary bound, 26 Nov 2013. 120,136

[Bel15] A. Belovs. Variations on quantum adversary. 2015. 102

[Ben82] P. A. Benioff. Quantum mechanical Hamiltonian models of Turing machines. 29(3):515–546, 1982. 1

[BHMT02] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, pages 53–74. 2002. quant-ph/0005055. 30,41,88,89,111

[BHT97] G. Brassard, P. Høyer, and A. Tapp. Quantum algorithm for the collision problem. *ACM SIGACT News (Cryptology Column)*, 28:14–19, 1997. quant-ph/9705002. 31

[BJ99] N. H. Bshouty and J. C. Jackson. Learning DNF over the uniform distribution using a quantum example oracle. *SIAM Journal on Computing*, 28(3):1136–1153, 1999. Earlier version in COLT'95. 8,110,112,115,116,133,134,137,142

[BK02] H. Barnum and E. Knill. Reversing quantum dynamics with near-optimal quantum and classical fidelity. *Journal of Mathematical Physics*, 43:2097–2106, 2002. quant-ph/0004088. 147

[BKL+17] F. G. S. L. Brandão, A. Kalev, T. Li, C. Y. Lin, K. Svore, and X. Wu. Exponential quantum speed-ups for semidefinite programming with applications to quantum learning. arXiv:1710.02581, 2017. 84

[BKT17] M. Bun, R. Kothari, and J. Thaler. The polynomial method strikes back: Tight quantum query bounds via dual polynomials. arXiv:1710.09079, 2017. 55

[BL98] P. Bartlett and P. M. Long. Prediction, learning, uniform convergence, and scale-sensitive dimensions. *Journal of Computer and System Sciences*, 56(2):174–190, 1998. 129

[BMMN13] M. Braverman, K. Makarychev, Y. Makarychev, and A. Naor. The Grothendieck constant is strictly smaller than Krivine's bound. *Forum Math. Pi*, 1:453–462, 2013. Preliminary version in FOCS'11. arXiv:1103.6161. [59,75]

[BS17] F. G. S. L. Brandão and K. Svore. Quantum speed-ups for semidefinite programming. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, 2017. arXiv:1609.05537. [83,84,105]

[BT09] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. [89]

[Bul05a] D. Bulger. Quantum basin hopping with gradient-based local optimisation. Unpublished, 2005. [87,89]

[Bul05b] D. Bulger. Quantum computational gradient estimation. Unpublished, 2005. [91]

[BV97] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. Earlier version in STOC'93. [113,142]

[BV13] J. Briët and T. Vidick. Explicit lower and upper bounds on the entangled value of multiplayer XOR games. *Communications in Mathematical Physics*, 321(1):181–207, 2013. arXiv:1108.5647. [59]

[BVW07] H. Buhrman, N. K. Vereshchagin, and R. de Wolf. On computation and communication with small bias. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007)*, pages 24–32, 2007. [60]

[BW02] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002. [13]

[BWP+17] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. Quantum machine learning. *Nature*, 549:195–202, 2017. arXiv:1611.09347. [8,111]

[Chi11] A. Childs. Lecture notes on quantum algorithms, 2011. Technical report, University of Maryland. Available at https://cs.umd.edu/amchilds/qa/. [16,28]

[CHI+17] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig. Quantum machine learning: a classical perspective. arXiv:1707.08561, 2017. [8,111]

[CHTW04] R. Cleve, P. Høyer, B. Toner, and J. Watrous. Consequences and limits of nonlocal strategies. In *19th Annual IEEE Conference on Computational Complexity (CCC 2004)*, pages 236–249, 2004. arXiv:quant-ph/0404076. [59]

[CHY16] H. C. Cheng, M. H. Hsieh, and P. C. Yeh. The learnability of unknown quantum measurements. *Quantum Information and Computation*, 16(7&8):615–656, 2016. arXiv:1501.00559. [131]

[CKS15] A. M. Childs, R. Kothari, and R. D. Somma. Quantum linear systems algorithm with exponentially improved dependence on precision. arXiv:1511.02306, 2015. [83]

[CS87] E. Christensen and A. M. Sinclair. Representations of completely bounded multilinear operators. *Journal of Functional analysis*, 72(1):151–181, 1987. [58,64]

[CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. 1991. [146]

[Dav84] A. Davie. Lower bound for K_G. Unpublished, 1984. [75]

[DB17] V. Dunjko and H. J. Briegel. Machine learning & artificial intelligence in the quantum domain. arXiv: 1709.02779, 2017. [8,111]

[Deu85] D. Deutsch. Quantum theory, the Church-Turing principle, and the universal quantum Turing machine. In *Proceedings of the Royal Society of London*, volume A400, pages 97–117, 1985. [1,3]

[DH96] C. Dürr and P. Høyer. A quantum algorithm for finding the minimum. quant-ph/9607014, 18 Jul 1996. [31,87,89]

[DHHM06] C. Dürr, M. Heiligman, P. Høyer, and M. Mhalla. Quantum query complexity of some graph problems. *SIAM Journal on Computing*, 35(6):1310–1328, 2006. Earlier version in ICALP'04. [31]

[DJ92] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 439(1907), 1992. [24]

[DJT95] J. Diestel, H. Jarchow, and A. Tonge. *Absolutely summing operators*, volume 43 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1995. [59,70]

[Dör07] S. Dörn. *Quantum Complexity of Graph and Algebraic Problems*. PhD thesis, Institut für Theoretische Informatik, 2007. [31]

[DS16] A. Daniely and S. Shalev-Shwartz. Complexity theoretic limitations on learning DNF's. In *Proceedings of the 29th Conference on Learning Theory (COLT'16)*, 2016. 137

[EF01] Y. C. Eldar and G. D. Forney Jr. On quantum detection and the square-root measurement. *IEEE Transactions and Information Theory*, 47(3):858–872, 2001. quant-ph/0005132. 149

[EHKV89] A. Ehrenfeucht, D. Haussler, M. J. Kearns, and L. G. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–261, 1989. Earlier version in COLT'88. 144

[EMV03] Y. C. Eldar, A. Megretski, and G. C. Verghese. Designing optimal quantum detectors via semidefinite programming. *IEEE Transactions Information Theory*, 49(4):1007–1012, 2003. quant-ph/0205178. 148

[EZ64] H. Ehlich and K. Zeller. Schwankung von Polynomen zwischen Gitterpunkten. *Mathematische Zeitschrift*, 86:41–44, 1964. 35

[Fey82] R. Feynman. Simulating physics with computers. 21(6/7):467–488, 1982. 1

[Fey85] R. Feynman. Quantum mechanical computers. *Optics News*, 11:11–20, 1985. 1

[FGG14] E. Farhi, J. Goldstone, and S. Gutmann. A quantum approximate optimization algorithm. arXiv:1411.4028, 2014. 6,84,102

[FGGS00] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum computation by adiabatic evolution. quant-ph/0001106, 2000. 6,84

[Fre95] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995. Earlier version in COLT'90. 133

[FSA99] Y. Freund, R. Schapire, and N. Abe. A short introduction to boosting. *Journal of Japanese Society For Artificial Intelligence*, 14:771–780, 1999. 105

[Gar84] L. T. Gardner. An elementary proof of the Russo-Dye theorem. *Proceedings of the American Mathematical Society*, 90(1):171, 1984. 63

[Gav12] D. Gavinsky. Quantum predictive learning and communication complexity with single input. *Quantum Information & Computation*, 12(7-8):575–588, 2012. Earlier version in COLT'10. arXiv:0812.3429. 127

[GAW17] A. Gilyén, S. Arunachalam, and N. Wiebe. Optimizing quantum optimization algorithms via faster quantum gradient computation. arXiv:1711.00465, 2017. *v*,83,85,91,98,99,101,102,103,104,111

[GH01] C. Gentile and D. P. Helmbold. Improved lower bounds for learning from noisy examples: An information-theoretic approach. *Information and Computation*, 166:133–155, 2001. 144

[GL89] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of 21st ACM STOC*, pages 25–32, 1989. 133

[GOS+11] P. Gopalan, R. O'Donnell, R. A. Servedio, A. Shpilka, and K. Wimmer. Testing Fourier dimensionality and sparsity. 40(4):1075–1100, 2011. Earlier version in ICALP'09. 136

[Got98] D. Gottesman. The heisenberg representation of quantum computers. arXiv:quant-ph/9807006, 1998. 131

[GPW15] M. Göös, T. Pitassi, and T. Watson. Deterministic communication vs. partition number. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS*, pages 1077–1088, 2015. 13,25

[Gro53] A. Grothendieck. Résumé de la théorie métrique des produits tensoriels topologiques (French). *Bol. Soc. Mat. São Paulo*, 8:1–79, 1953. 59,75,78

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996. 2,4,5,24,25,30,82,83,84,111,195,199

[Gro02] L. K. Grover. Trade-offs in the quantum search algorithm. 66(052314), 2002. quant-ph/0201152. 37,38,39,40,43,45

[Han16] S. Hanneke. The optimal sample complexity of PAC learning. *Journal of Machine Learning Research*, 17(38):1–15, 2016. arXiv:1507.00473. 9,125,126,141,149

[Hau92] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992. 9,114,116,141

[Heg95] T. Hegedűs. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the 8th Conference on Learning Theory COLT*, pages 108–117, 1995. 123,124,125

[HHJ+16] J. Haah, A. W. Harrow, Z. Ji, X. Wu, and N. Yi. Sample-optimal tomography of quantum states. In *Proceedings of 48th ACM STOC*, pages 913–925, 2016. arXiv:1508.01797. 128,130

[HHL09] A. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for solving linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. arXiv:0811.3171. 83,84,87,111

[HIKP12] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Nearly optimal sparse Fourier transform. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC*, pages 563–578, 2012. arXiv:1201.2501. 136

[HJS+96] P. Hausladen, R. Jozsa, B. Schumacher, M. Westmoreland, and W. Wootters. Classical information capacity of a quantum channel. *Physical Review A*, 54:1869, 1996. 145

[HLM17] A. Harrow, C. Y. Lin, and A. Montanaro. Sequential measurements, disturbance and property testing. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*, pages 1598–1611, 2017. arXiv:1607.03236. 131

[HLŠ07] P. Høyer, T. Lee, and R. Špalek. Negative weights make adversaries stronger. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, 2007*, pages 526–535, 2007. arXiv:quant-ph/0611054. 5,25,26,28,196,199

[HMP+10] M. Hunziker, D. A. Meyer, J. Park, J. Pommersheim, and M. Rothstein. The geometry of quantum learning. *Quantum Information Processing*, 9(3):321–341, 2010. quant-ph/0309059. 124

[Hol73] A. S. Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problemy Peredachi Informatsii*, 9(3):3–11, 1973. English translation in *Problems of Information Transmission*, 9:177–183, 1973. 119

[HP00] T. Hogg and D. Portnov. Quantum optimization. *Information Sciences*, 128(3-4):181–197, 2000. arXiv:quant-ph/0006090. 6,84

[HR16] I. Haviv and O. Regev. The list-decoding size of Fourier-sparse Boolean functions. *ACM Transactions on Computation Theory*, 8(3):10:1–10:14, 2016. Earlier version in CCC'15. arXiv:1504.01649. 136,137

[HW94] P. Hausladen and W.K. Wootters. A pretty good measurement for distinguishing quantum states. *Journal of Modern Optics*, 41(12):2385–2390, 1994. 145

[Jac97] J. C. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997. Earlier version in FOCS'94. 133,134

[JKK+17] P. Jain, M. S. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford. Accelerating stochastic gradient descent. arXiv:1704.08227, 2017. 89

[JKP09] N. Johnston, D. W. Kribs, and V. I. Paulsen. Computing stabilized norms for quantum operations via the theory of completely bounded maps. *Quantum Information & Computation*, 9(1):16–35, 2009. arXiv:0711.3636. 64

[Jor05] S. P. Jordan. Fast quantum algorithm for numerical gradient estimation. *Physical Review Letters*, 95(5):050501, 2005. 83,85,86,87,88,91,92,93,97,100

[Jor08] S. P. Jordan. *Quantum Computation Beyond the Circuit Model*. PhD thesis, Massachusetts Institute of Technology, 2008. 105

[JTY02] J. C. Jackson, C. Tamon, and T. Yamakami. Quantum DNF learnability revisited. In *Proceedings of 8th COCOON*, pages 595–604, 2002. quant-ph/0202066. 134

[JZ09] R. Jain and S. Zhang. New bounds on classical and quantum one-way communication complexity. *Theoretical Computer Science*, 410(26):2463–2477, 2009. arXiv:0802.4101. 152

[Kim13] S. Kimmel. Quantum adversary (upper) bound. *Chicago Journal of Theoretical Computer Science*, 2013(4), 2013. Earlier version in ICALP'12. arXiv:1101.0797. 81

[KLM06] P. Kaye, R. Laflamme, and M. Mosca. *An Introduction to Quantum Computing*. Oxford University Press, 2006. 16,28,150

[KLW15] J. Kaniewski, T. Lee, and R. de Wolf. Query complexity in expectation. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP*, pages 761–772, 2015. arXiv:1411.7280. 60

[KN98] T. Kadowaki and H. Nishimori. Quantum annealing in the transverse Ising model. *Physical Review E*, 58(5), 1998. cond-mat/9804280. [6,84]

[KN12] S. Khot and A. Naor. Grothendieck-type inequalities in combinatorial optimization. *Communications on Pure and Applied Mathematics*, 65(7):992–1035, 2012. arXiv:1108.2464. [59]

[Kot12] R. Kothari. Quantum computing and learning theory. Unpublished manuscript, 2012. [111]

[Kot14] R. Kothari. An optimal quantum algorithm for the oracle identification problem. In *31st International Symposium on Theoretical Aspects of Computer Science STACS*, pages 482–493, 2014. arXiv:1311.7685. [31,113,121,122,124]

[KP16] A. Kontorovich and I. Pinelis. Exact lower bounds for the agnostic probably-approximately-correct (PAC) machine learning model, 2016. Preprint at arXiv:1606.08920. [144]

[KP17a] I. Kerenidis and A. Prakash. Quantum gradient descent for linear systems and least squares. arXiv:1704.04992, 2017. [87,89]

[KP17b] I. Kerenidis and A. Prakash. Quantum recommendation systems. In *Innovations in Theoretical Computer Science ITCS*, 2017. arXiv:1603.08675. [111]

[KS04] A. Klivans and R. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. *Journal of Computer and System Sciences*, 68(2):303–318, 2004. Earlier version in STOC'01. [137]

[KSS94] M. J. Kearns, R. E. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994. Earlier version in COLT'92. [9,114,116,141]

[KV94a] M. J. Kearns and L. G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994. [132]

[KV94b] M. J. Kearns and U. V. Vazirani. *An introduction to computational learning theory*. MIT Press, 1994. [110,115]

[Li05] J. Li. General explicit difference formulas for numerical differentiation. *Journal of Computational and Applied Mathematics*, 183(1):29–52, 2005. [98]

[LL16] C. Y. Lin and H. H. Lin. Upper bounds on quantum query complexity inspired by the elitzur–vaidman bomb tester. *Theory of Computing*, 12(1):1–35, 2016. Earlier version in CCC'16. arXiv:1410.0932. [25,31,81,113]

[LLV15] C. M. Le, E. Levina, and R. Vershynin. Sparse random graphs: regularization and concentration of the Laplacian. 2015. arXiv:1502.03049. [60]

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM*, 40(3):607–620, 1993. Earlier version in FOCS'89. [137]

[LMP15] T. Laarhoven, M. Mosca, and J. van de Pol. Finding shortest lattice vectors faster using quantum search. *Designs, Codes and Cryptography*, 77(2-3):375–400, 2015. [31]

[LMR+11] T. Lee, R. Mittal, B. W. Reichardt, R. Špalek, and M. Szegedy. Quantum query complexity of state conversion. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 344–353, 2011. arXiv:1011.3020. [5,26,28,81]

[LMR13a] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum algorithms for supervised and unsupervised machine learning, 1 Jul 2013. arXiv:1307.0411. [111]

[LMR13b] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(631–633), 2013. arXiv:1307.0401. [111]

[LPL14] P. C. S. Lara, R. Portugal, and C. Lavor. A new hybrid classical-quantum algorithm for continuous global optimization problems. *Journal on Global Optimization*, 60(2):317–331, 2014. [89]

[Man80] Y. Manin. Vychislimoe i nevychislimoe (computable and noncomputable). *Soviet Radio*, pages 13–15, 1980. In Russian. [1]

[Man99] Y. Manin. Classical computing, quantum computing, and Shor's factoring algorithm. quant-ph/9903008, 2 Mar 1999. [1]

[MNR11] A. Montanaro, H. Nishimura, and R. Raymond. Unbounded-error quantum query complexity. *Theoretical Computer Science*, 412(35):4619–4628, 2011. arXiv:0712.1446. [60]

[MNRS11] F. Magniez, A. Nayak, J. Roland, and M. Santha. Search via quantum walk. *SIAM Journal on Computing*, 40(1):142–164, 2011. Earlier version in STOC'07. arXiv:quant-ph/0608026. [25]

[Mon07] A. Montanaro. On the distinguishability of random quantum states. *Communications in Mathematical Physics*, 273(3):619–636, 2007. quant-ph/0607011. [148]

[Mon15] A. Montanaro. Quantum speedup of Monte Carlo methods. *Proceedings of Royal Society A*, 471(2181), 2015. arXiv:1504.06987. [6,84]

[Mon17a] A. Montanaro. Learning stabilizer states by bell sampling, 2017. arXiv:1707.04012. [131]

[Mon17b] A. Montanaro. Quantum pattern matching fast on average. *Algorithmica*, 77(1):16–39, 2017. arXiv:1408.1816. [31]

[Mos83] M. Yu. Moshkov. Conditional tests. *Problemy Kibernetzkt*, 40:131–170, 1983. In Russian. [124,125]

[MOS04] E. Mossel, R. O'Donnell, and R. Servedio. Learning functions of k relevant variables. *Journal of Computer and System Sciences*, 69(3):421–434, 2004. Earlier version in STOC'03. [135]

[Mur14] G. J. Murphy. *C^*-algebras and operator theory*. Academic press. 2014. [62]

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000. [16]

[NC02] M. A. Nielsen and I. Chuang. Quantum computation and quantum information, 2002. [96,146]

[Nes83] Y. Nesterov. A method for solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. *Doklady Akademii Nauk SSSR*, 269:543–547, 1983. [89]

[Nis91] N. Nisan. Crew prams and decision trees. *SIAM Journal on Computing*, 20(6):999–1007, 1991. [25]

[NS94] N. Nisan and M. Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity*, 4:301–313, 1994. Earlier version in STOC'92. [35]

[O'D14] R. O'Donnell. *Analysis of Boolean Functions*. Cambridge university Press, 2014. [15]

[OP99] T. Oikhberg and G. Pisier. The "maximal" tensor product of operator spaces. *Proceedings of the Edinburgh Mathematical Society*, 42(2):267–284, 1999. [57]

[OW16] R. O'Donnell and J. Wright. Efficient quantum tomography. In *Proceedings of 48th ACM STOC*, pages 899–912, 2016. arXiv:1508.01907. [128,130]

[Pau02] V. Paulsen. *Completely bounded maps and operator algebras*, volume 78. Cambridge University Press, Cambridge, 2002. [62,64]

[PGWP+08] D. Pérez-García, M. Wolf, C. Palazuelos, I. Villanueva, and M. Junge. Unbounded violation of tripartite Bell inequalities. *Communications in Mathematical Physics*, 279:455, 2008. arXiv:quant-ph/0702189. [59,60]

[Pis03] G. Pisier. *Introduction to operator space theory*, volume 294 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2003. [58,64]

[Pis12] G. Pisier. Grothendieck's theorem, past and present. *Bull. Amer. Math. Soc.*, 49(2):237–323, 2012. also available at arXiv:1101.4195. [59]

[PMS+14] A. Peruzzo, J. McClean, P. Shadbolt, M.H. Yung, X.Q. Zhou, P. J. Love, A.G. Alán, and J.L. Obrien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5, 2014. arXiv:1304.3061. [102]

[Pol12] D. Pollard. *Convergence of stochastic processes*. Science & Business Media. Springer, 2012. [70]

[PS87] V. I. Paulsen and R. R. Smith. Multilinear maps and tensor norms on operator systems. *Journal of functional analysis*, 73(2):258–276, 1987. [58]

[PV16] C. Palazuelos and T. Vidick. Survey on nonlocal games and operator space theory. *Journal of Mathematical Physics*, 57(1):015220, 2016. [60]

[RC66] T. J. Rivlin and E. W. Cheney. A comparison of uniform approximations on an interval and a finite subset thereof. *SIAM Journal on Numerical Analysis*, 3(2):311–320, 1966. [35]

[Ree91] J. Reeds. A new lower bound on the real Grothendieck constant. Manuscript (`http://www.dtc.umn.edu/~reedsj/bound2.dvi`), 1991. [59,75]

[Rei09] B. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function.

In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009*, pages 544–551, 2009. arXiv:0904.2759. 5,25,26,28,81

[Rei11] B. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011*, pages 560–569, 2011. arXiv:1005.1601. 5,26,28,82

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, pages 318–362. 1986. 103

[RML13] P. Rebentrost, M. Mohseni, and S. Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(13), 2013. arXiv:1307.0471. 111

[ROA16] J. Romero, J. P. Olson, and A. G. Alan. Quantum autoencoders for efficient compression of quantum data. arXiv:1612.02806, 2016. 103,104

[Roc17] A. Rocchetto. Stabiliser states are efficiently PAC-learnable, 2017. arXiv:1705.00345. 131

[RSPL16] P. Rebentrost, M. Schuld, F. Petruccione, and S. Lloyd. Quantum gradient descent and Newton's method for constrained polynomial optimization. arXiv:1612.01789, 2016. 87,89

[Rud91] W. Rudin. *Functional analysis*. McGraw-Hill Science, 1991. 79

[Rud16] S. Ruder. An overview of gradient descent optimization algorithms. arXiv: 1609.04747, 2016. 86

[SB14] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014. 110,115,127,142,144,171

[SG04] R. Servedio and S. Gortler. Equivalences and separations between quantum and classical learnability. *SIAM Journal on Computing*, 33(5):1067–1092, 2004. Combines earlier papers from ICALP'01 and CCC'01. quant-ph/0007036. 8,110,119,120,124,132,133,143

[She13] A. A. Sherstov. Making polynomials robust to noise. *Theory of Computing*, 9:593–615, 2013. Earlier version in STOC'12. 82

[Sho97] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. Earlier version in FOCS'94 and quant-ph/9508027. 1,4,24,30,84,132

[Sim96] H. U. Simon. General bounds on the number of examples needed for learning probabilistic concepts. *Journal of Computer and System Sciences*, 52(2):239–254, 1996. Earlier version in COLT'93. 127,142,146,149

[Sim97] D. Simon. On the power of quantum computation. *SIAM journal of computing*, 26(5):1474–1483, 1997. Earlier version in FOCS'94. 4,25,133

[Sim15] H. U. Simon. An almost optimal PAC algorithm. In *Proceedings of the 28th Conference on Learning Theory (COLT)*, pages 1552–1563, 2015. 125,141

[Smi88] R. R. Smith. Completely bounded multilinear maps and Grothendieck's inequality. *Bulletin of the London Mathematical Society*, 20(6):606–612, 1988. 58

[ŠS06] R. Špalek and M. Szegedy. All quantum adversary methods are equivalent. *Theory of Computing*, 2(1):1–18, 2006. Earlier version in ICALP'05. quant-ph/0409116. 28

[SSP15] M. Schuld, I. Sinayskiy, and F. Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015. arXiv:1409.3097. 8,111

[Tal94] M. Talagrand. Sharper bounds for Gaussian and empirical processes. *The Annals of Probability*, pages 28–76, 1994. 127,142,149

[Tao12] T. Tao. *Topics in random matrix theory*, volume 132. American Mathematical Society, 2012. 70

[TCR10] M. Tomamichel, R. Colbeck, and R. Renner. Duality between smooth min- and max-entropies. *IEEE Transactions on Information Theory*, 56(9):4674–4681, 2010. arXiv:0907.5238. 104

[Tho14] E. Thomas. A polarization identity for multilinear maps. *Indagationes Mathematicae*, 25:468–474, 2014. arXiv:1309.1275. 74

[Tro09] J. A. Tropp. Column subset selection, matrix factorization, and eigenvalue optimization. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 978–986, 2009. 60

[Tsi87] B. S. Tsirelson. Quantum analogues of the Bell inequalities. The case of two spatially separated domains. *J. Soviet Math.*, 36:557–570, 1987. [59]

[Val84] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. [2,110,114,115,140]

[VC71] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971. [8,125]

[VC74] V. Vapnik and A. Chervonenkis. Theory of pattern recognition. 1974. In Russian. [127,142,149]

[Ver90] K. A. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory (COLT'90)*, pages 314–326, 1990. [133]

[Wat11] J. Watrous. Theory of quantum information. Lecture notes, 2011. [16,146]

[WBL12] N. Wiebe, D. Braun, and S. Lloyd. Quantum algorithm for data fitting. *Physical review letters*, 109(5):050505, 2012. [6,84]

[WDK+16] K. H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, and M. S. Kim. Quantum generalisation of feedforward neural networks. arXiv:1612.01045, 2016. [103]

[WHT15] D. Wecker, M. B. Hastings, and M. Troyer. Progress towards practical quantum variational algorithms. *Physical Review A*, 92(4):042303, 2015. arXiv:1507.08969. [102]

[Wit14] P. Wittek. *Quantum Machine Learning: What Quantum Computing Means to Data Mining*. Elsevier, 2014. [8,111]

[WKS15] N. Wiebe, A. Kapoor, and K. Svore. Quantum nearest-neighbor algorithms for machine learning. *Quantum Information and Computation*, 15(3&4):318–358, 2015. [84]

[WKS16a] N. Wiebe, A. Kapoor, and K. Svore. Quantum deep learning. *Quantum Information and Computation*, 16(7):541–587, 2016. arXiv:1412.3489. [111]

[WKS16b] N. Wiebe, A. Kapoor, and K. Svore. Quantum perceptron models. In *Neural Information Processing Systems (NIPS)*, pages 3999–4007, 2016. arXiv: 1602.04799. [84,111]

[Wol03] R. de Wolf. Nondeterministic quantum query and communication complexities. *SIAM Journal on Computing*, 32(3):681–699, 2003. cs.CC/0001014. 60

[Wol08] R. de Wolf. A brief introduction to Fourier analysis on the Boolean cube. *Theory of Computing, Graduate Surveys*, 1:1–20, 2008. 15

[Wol10] R. de Wolf. A note on quantum algorithms and the minimal degree of ϵ-error polynomials for symmetric functions. *Quantum Information & Computation*, 8(10):943–950, 2010. arXiv:0802.1816. 30

[Wol13] R. de Wolf. Quantum computing: Lecture notes, 2013. Lecture notes. Available at https://homepages.cwi.nl/ rdewolf/qcnotes.pdf. 16,97

[Yao77] A. C-C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 222–227, 1977. 121

[Zal99] C. Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A*, 60:2746–2751, 1999. quant-ph/9711070. 30,34,38

[Zha05] S. Zhang. On the power of Ambainis lower bounds. *Theoretical Computer Science*, 339(2):241256, 2005. arXiv:quant-ph/0311060. 28

[Zha10] C. Zhang. An improved lower bound on query complexity for quantum PAC learning. *Information Processing Letters*, 111(1):40–45, 2010. 126,143

Abstract

In this thesis, we present results in two directions of research. In the first part we study query and gate complexity of quantum algorithms for certain problems and in the second part we study sample and query complexity of quantum machine learning algorithms.

Part I: Quantum algorithms

In the first part we present three contributions to quantum algorithms, which we briefly summarize below.

Chapter 3. We look at the following basic search problem: suppose there is an unstructured database consisting of N elements and one of the elements is "marked". Our goal is to find the marked element. To solve this problem we are allowed to make queries which tell us if an element is marked or not. Ideally we would like to find the marked element making as few queries as possible. Classically, in the worst case one would need to make essentially N queries to find the marked element.

Grover [Gro96] constructed a quantum algorithm that solves this problem using $O(\sqrt{N})$ quantum queries and $O(\sqrt{N}\log N)$ other elementary gates. It is known that the number of quantum queries necessary to solve the search problem is $\Omega(\sqrt{N})$, so Grover's algorithm cannot be improved in terms of queries. In this chapter we describe a new quantum algorithm to solve the search problem, whose gate complexity is essentially $O(\sqrt{N})$, while preserving the query complexity of Grover's algorithm.

Chapter 4. The flip-side of obtaining new quantum algorithms is showing query lower bounds, i.e., showing that every quantum algorithm needs to make at least a certain number of queries in order to solve a problem. In this direction there are two famous techniques to give query lower bounds, the polynomial

method [BBC+01] and the adversary method [Amb00, HLŠ07]. The adversary method is known to characterize quantum query complexity, i.e., one can obtain upper bounds on quantum query complexity using the adversary method.

A natural question is whether the polynomial method admits such a converse as well. In this chapter we give a positive answer to this question by introducing a new degree-measure called the *completely bounded approximate degree* (denoted cb-deg) of a Boolean function. We show that for a Boolean function f, this cb-deg(f) equals the quantum query complexity of f. Our succinct characterization of quantum algorithms in terms of polynomials not only refines the polynomial method, but it also gives a new technique for showing upper and lower bounds on quantum query complexity.

Chapter 5. Optimization is an important task that touches on virtually every area of science. For $f : \mathbb{R}^d \to \mathbb{R}$, consider the following optimization problem: $\mathrm{OPT} = \min\{f(x) : x \in \mathbb{R}^d\}$. One generic technique used often to compute OPT is the gradient-descent algorithm. This begins with an arbitrary $\boldsymbol{x} \in \mathbb{R}^d$, computes the gradient of f at $\boldsymbol{x}$ (denoted $\nabla f(\boldsymbol{x})$) and moves to a point $\boldsymbol{x}'$ in the direction of $-\nabla f(\boldsymbol{x})$. This process is repeated a few times before the algorithm hopefully reaches a good approximation of OPT. Given the simplicity and generality of the algorithm, gradient-based methods are ubiquitous in machine learning algorithms.

An integral part of the gradient-based algorithm is the gradient computation step. Can the gradient computation step be improved using quantum techniques? In this chapter we develop a quantum algorithm that calculates the gradient of $f : \mathbb{R}^d \to \mathbb{R}$ quadratically faster than classical methods. To be precise, we show that in order to obtain a ε-coordinate-wise approximation of the d-dimensional gradient vector ∇f at a given point $\boldsymbol{x}$, it suffices to make $O(\sqrt{d}/\varepsilon)$ queries to the oracle encoding f. Using our quantum gradient calculation algorithm coupled with other quantum subroutines, we provide a quadratic quantum improvement to the complexity of almost all gradient-based optimization algorithms.

Part II: Quantum learning theory

In the second part we present two contributions to quantum learning theory, which we briefly summarize below.

Chapter 6. In the last decade, with the explosion of data and information, machine learning has gained prominence. Alongside the boom in classical machine learning, the last few years have seen an increase in the interest in *quantum machine learning*, an interdisciplinary area that uses the powers of quantum physics to improve classical machine learning algorithms.

In this chapter, we survey the theoretical side of quantum machine learning: quantum learning theory. We describe the main results known for three models

of learning, using classical as well as quantum data: exact learning from membership queries, the probably approximately correct (PAC) learning model and the agnostic learning model. Apart from information-theoretic results, we also survey results on the time complexity of learning from membership queries and learning in the PAC and agnostic models.

Chapter 7. Leslie Valiant's PAC model of learning gives a complexity-theoretic definition of what it means for a concept class $\mathcal{C}$ (i.e., a collection of Boolean functions) to be (efficiently) learnable. In the PAC model, our goal is to approximately learn an unknown Boolean function from $\mathcal{C}$ given random examples for that function. It is well-known that the number of random examples necessary and sufficient to PAC-learn $\mathcal{C}$ is given by a combinatorial parameter, the VC dimension of $\mathcal{C}$.

In this chapter we ask if a *quantum* learner can PAC-learn $\mathcal{C}$ given fewer quantum examples. We give a negative answer by showing that the number of quantum examples necessary and sufficient to PAC-learn $\mathcal{C}$ is also given by the VC dimension of $\mathcal{C}$. We consider more realistic and flexible versions of PAC learning, i.e., agnostic learning and learning under random classification noise. In both these learning models, we show that quantum examples are not more powerful than classical examples.

Samenvatting

In dit proefschrift worden resultaten gepresenteerd in twee onderzoeksrichtingen. In het eerste deel bestuderen we query en gate-complexiteit van quantumalgoritmes voor bepaalde problemen. Het tweede deel omvat sample en query-complexiteit van quantum machine learning algoritmes.

Deel I: Quantum algoritmes

In het eerste deel behandelen we drie bijdragen aan quantumalgoritmes, die we hier kort samenvatten.

Hoofdstuk 3. We bekijken het volgende simpele zoekprobleem: er is een ongestructureerde database van N elementen en één van de elementen is "gemarkeerd". Het doel is om het gemarkeerde element te vinden. Om dit op te lossen mogen we queries doen die ons vertellen of een element gemarkeerd is, en we willen graag zo min mogelijk van deze queries doen. Voor een klassiek algoritme kost het in het ergste geval N queries om het gemarkeerde element te vinden.

Grover [Gro96] heeft een quantumalgoritme bedacht dat dit probleem oplost met $O(\sqrt{N})$ quantum queries en $O(\sqrt{N}\log N)$ andere elementaire gates. Het is bekend dat het aantal benodigde queries om dit probleem op te lossen $\Omega(\sqrt{N})$ is, dus Grovers algoritme kan niet worden verbeterd wat betreft het aantal queries. In dit hoofdstuk beschrijven we een nieuw quantumalgoritme om dit zoekprobleem op te lossen met een gate-complexiteit van ongeveer $O(\sqrt{N})$ en met dezelfde query-complexiteit als Grovers algoritme.

Hoofdstuk 4. Naast quantumalgoritmes zijn er quantum query-ondergrenzen. Deze laten zien dat elk quantum algoritme minimaal een bepaald aantal queries moet doen om een probleem op te lossen. In deze richting bestaan twee bekende technieken om ondergrenzen aan te tonen. Dit zijn de polynoom-methode [BBC+01] en de adversary methode [Amb00, HLŠ07]. De adversary methode

staat bekend om zijn karakterisatie van quantum query-complexiteit omdat het ook bovengrenzen geeft op het aantal benodigde queries.

Een natuurlijke vraag is of de polynoom-methode ook bovengrenzen kan geven. In dit hoofdstuk geven we een positief antwoord op deze vraag door het introduceren van de zogeheten "completely bounded approximate degree" (afgekort cb-deg) van een Booleaanse functie. We laten zien dat voor een Booleaanse functie f, de cb-deg(f) gelijk is aan de quantum query-complexiteit van f. Onze beknopte karakterisatie van quantumalgoritmes in termen van polynomen verfijnt niet alleen de polynoommethode maar geeft ook nieuwe technieken om onder- en bovengrenzen te bepalen voor query-complexiteit.

Hoofdstuk 5. Optimalisatie is in praktisch alle gebieden van de wetenschap belangrijk. Laat $f : \mathbb{R}^d \to \mathbb{R}$ en beschouw het optimalisatieprobleem $\mathrm{OPT} = \min\{f(x) : x \in \mathbb{R}^d\}$. Een generieke techniek om OPT uit te rekenen is het gradient descent algoritme. In dit algoritme wordt de gradiënt van f in $\boldsymbol{x}$ (uitgerekend $\nabla f(\boldsymbol{x})$) om zo naar een punt $\boldsymbol{x}'$ in de richting van $-\nabla f(\boldsymbol{x})$ te gaan. Dit wordt een aantal keer herhaald om hopelijk een goede benadering van OPT te vinden. Vanwege de eenvoud en algemeenheid van dit algoritme komen gradient descent methoden overal in machine learning algoritmes voor.

Een belangrijke stap van dit algoritme is de berekening van de gradiënt. Kan deze berekening worden versneld met quantum technieken? In dit hoofdstuk ontwikkelen we een quantumalgoritme dat de gradiënt van f kwadratisch sneller uitrekent dan de klassieke methode. Om in elke coördinaat een ε-benadering te verkrijgen van de d-dimensionale gradiëntvector ∇f in $\boldsymbol{x}$, voldoet het om $O(\sqrt{d}/\varepsilon)$ queries te doen naar de functie f. Door ons quantum gradiëntalgoritme met andere quantumalgoritmes te combineren verkrijgen we een kwadratische verbetering voor de complexiteit van bijna alle gradient descent optimalisatiealgoritmes.

Deel II: Quantum learning theorie

In het tweede deel geven we twee bijdragen aan de quantum learning theorie, hier kort samengevat.

Hoofdstuk 6. Door de explosieve toename van de beschikbaarheid van data is machine learning erg groot geworden. Daarnaast is er in de laatste paar jaar ook interesse ontstaan voor *quantum* machine learning, een interdisciplinair vakgebied dat de kracht van quantummechanica gebruikt om machine learning algoritmes te verbeteren.

In dit hoofdstuk geven we een overzicht van de theoretische kant van quantum machine learning: de quantum leertheorie. We geven de voornaamste resultaten die bekend zijn voor drie modellen van learning voor zowel klassieke als quantum data: exact leren via membership queries, het probably approximately correct (PAC) model, en het agnostisch leren model. Naast de informatietheoretische

resultaten beschouwen we ook resultaten over de tijdcomplexiteit van het leren in deze modellen.

Hoofdstuk 7. Het PAC model van Leslie Valiant geeft een complexiteitstheoretische definitie van wat het voor een conceptklasse $\mathcal{C}$ (een verzameling Booleaanse functies) betekent om efficiënt leerbaar te zijn. In het PAC model is het doel om een benadering van een onbekende functie uit $\mathcal{C}$ te leren door het bekijken van willekeurige voorbeelden van deze functie. Het is bekend dat het aantal voorbeelden dat noodzakelijk en voldoende is om $\mathcal{C}$ te PAC-leren gegeven wordt door een combinatorische parameter, de VC-dimensie van $\mathcal{C}$.

In dit hoofdstuk stellen we de vraag of het quantum PAC-leren van $\mathcal{C}$ met minder quantum-voorbeelden kan. We geven een negatief antwoord door te laten zien dat het aantal benodigde samples in dit geval ook gelijk is aan de VC-dimensie van $\mathcal{C}$. We beschouwen realistischere en flexibelere versies van PAC-leren zoals agnostisch leren en leren met ruis. In beide modellen laten we zien dat quantumvoorbeelden niet krachtiger zijn dan klassieke samples.

Titles in the ILLC Dissertation Series:

ILLC DS-2009-01: **Jakub Szymanik**
Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language

ILLC DS-2009-02: **Hartmut Fitz**
Neural Syntax

ILLC DS-2009-03: **Brian Thomas Semmes**
A Game for the Borel Functions

ILLC DS-2009-04: **Sara L. Uckelman**
Modalities in Medieval Logic

ILLC DS-2009-05: **Andreas Witzel**
Knowledge and Games: Theory and Implementation

ILLC DS-2009-06: **Chantal Bax**
Subjectivity after Wittgenstein. Wittgenstein's embodied and embedded subject and the debate about the death of man.

ILLC DS-2009-07: **Kata Balogh**
Theme with Variations. A Context-based Analysis of Focus

ILLC DS-2009-08: **Tomohiro Hoshi**
Epistemic Dynamics and Protocol Information

ILLC DS-2009-09: **Olivia Ladinig**
Temporal expectations and their violations

ILLC DS-2009-10: **Tikitu de Jager**
"Now that you mention it, I wonder. . . ": Awareness, Attention, Assumption

ILLC DS-2009-11: **Michael Franke**
Signal to Act: Game Theory in Pragmatics

ILLC DS-2009-12: **Joel Uckelman**
More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains

ILLC DS-2009-13: **Stefan Bold**
Cardinals as Ultrapowers. A Canonical Measure Analysis under the Axiom of Determinacy.

ILLC DS-2010-01: **Reut Tsarfaty**
Relational-Realizational Parsing

ILLC DS-2010-02: **Jonathan Zvesper**
Playing with Information

ILLC DS-2010-03: **Cédric Dégremont**
The Temporal Mind. Observations on the logic of belief change in interactive systems

ILLC DS-2010-04: **Daisuke Ikegami**
Games in Set Theory and Logic

ILLC DS-2010-05: **Jarmo Kontinen**
Coherence and Complexity in Fragments of Dependence Logic

ILLC DS-2010-06: **Yanjing Wang**
Epistemic Modelling and Protocol Dynamics

ILLC DS-2010-07: **Marc Staudacher**
Use theories of meaning between conventions and social norms

ILLC DS-2010-08: **Amélie Gheerbrant**
Fixed-Point Logics on Trees

ILLC DS-2010-09: **Gaëlle Fontaine**
Modal Fixpoint Logic: Some Model Theoretic Questions

ILLC DS-2010-10: **Jacob Vosmaer**
Logic, Algebra and Topology. Investigations into canonical extensions, duality theory and point-free topology.

ILLC DS-2010-11: **Nina Gierasimczuk**
Knowing One's Limits. Logical Analysis of Inductive Inference

ILLC DS-2010-12: **Martin Mose Bentzen**
Stit, Iit, and Deontic Logic for Action Types

ILLC DS-2011-01: **Wouter M. Koolen**
Combining Strategies Efficiently: High-Quality Decisions from Conflicting Advice

ILLC DS-2011-02: **Fernando Raymundo Velazquez-Quesada**
Small steps in dynamics of information

ILLC DS-2011-03: **Marijn Koolen**
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

ILLC DS-2011-04: **Junte Zhang**
System Evaluation of Archival Description and Access

ILLC DS-2011-05: **Lauri Keskinen**
Characterizing All Models in Infinite Cardinalities

ILLC DS-2011-06: **Rianne Kaptein**
Effective Focused Retrieval by Exploiting Query Context and Document Structure

ILLC DS-2011-07: **Jop Briët**
Grothendieck Inequalities, Nonlocal Games and Optimization

ILLC DS-2011-08: **Stefan Minica**
Dynamic Logic of Questions

ILLC DS-2011-09: **Raul Andres Leal**
Modalities Through the Looking Glass: A study on coalgebraic modal logic and their applications

ILLC DS-2011-10: **Lena Kurzen**
Complexity in Interaction

ILLC DS-2011-11: **Gideon Borensztajn**
The neural basis of structure in language

ILLC DS-2012-01: **Federico Sangati**
Decomposing and Regenerating Syntactic Trees

ILLC DS-2012-02: **Markos Mylonakis**
Learning the Latent Structure of Translation

ILLC DS-2012-03: **Edgar José Andrade Lotero**
Models of Language: Towards a practice-based account of information in natural language

ILLC DS-2012-04: **Yurii Khomskii**
Regularity Properties and Definability in the Real Number Continuum: idealized forcing, polarized partitions, Hausdorff gaps and mad families in the projective hierarchy.

ILLC DS-2012-05: **David García Soriano**
Query-Efficient Computation in Property Testing and Learning Theory

ILLC DS-2012-06: **Dimitris Gakis**
Contextual Metaphilosophy - The Case of Wittgenstein

ILLC DS-2012-07: **Pietro Galliani**
The Dynamics of Imperfect Information

ILLC DS-2012-08: **Umberto Grandi**
Binary Aggregation with Integrity Constraints

ILLC DS-2012-09: **Wesley Halcrow Holliday**
Knowing What Follows: Epistemic Closure and Epistemic Logic

ILLC DS-2012-10: **Jeremy Meyers**
Locations, Bodies, and Sets: A model theoretic investigation into nominalistic mereologies

ILLC DS-2012-11: **Floor Sietsma**
Logics of Communication and Knowledge

ILLC DS-2012-12: **Joris Dormans**
Engineering emergence: applied theory for game design

ILLC DS-2013-01: **Simon Pauw**
Size Matters: Grounding Quantifiers in Spatial Perception

ILLC DS-2013-02: **Virginie Fiutek**
Playing with Knowledge and Belief

ILLC DS-2013-03: **Giannicola Scarpa**
Quantum entanglement in non-local games, graph parameters and zero-error information theory

ILLC DS-2014-01: **Machiel Keestra**
Sculpting the Space of Actions. Explaining Human Action by Integrating Intentions and Mechanisms

ILLC DS-2014-02: **Thomas Icard**
The Algorithmic Mind: A Study of Inference in Action

ILLC DS-2014-03: **Harald A. Bastiaanse**
Very, Many, Small, Penguins

ILLC DS-2014-04: **Ben Rodenhäuser**
A Matter of Trust: Dynamic Attitudes in Epistemic Logic

ILLC DS-2015-01: **María Inés Crespo**
Affecting Meaning. Subjectivity and evaluativity in gradable adjectives.

ILLC DS-2015-02: **Mathias Winther Madsen**
The Kid, the Clerk, and the Gambler - Critical Studies in Statistics and Cognitive Science

ILLC DS-2015-03: **Shengyang Zhong**
Orthogonality and Quantum Geometry: Towards a Relational Reconstruction of Quantum Theory

ILLC DS-2015-04: **Sumit Sourabh**
Correspondence and Canonicity in Non-Classical Logic

ILLC DS-2015-05: **Facundo Carreiro**
Fragments of Fixpoint Logics: Automata and Expressiveness

ILLC DS-2016-01: **Ivano A. Ciardelli**
Questions in Logic

ILLC DS-2016-02: **Zoé Christoff**
Dynamic Logics of Networks: Information Flow and the Spread of Opinion

ILLC DS-2016-03: **Fleur Leonie Bouwer**
What do we need to hear a beat? The influence of attention, musical abilities, and accents on the perception of metrical rhythm

ILLC DS-2016-04: **Johannes Marti**
Interpreting Linguistic Behavior with Possible World Models

ILLC DS-2016-05: **Phong Lê**
Learning Vector Representations for Sentences - The Recursive Deep Learning Approach

ILLC DS-2016-06: **Gideon Maillette de Buy Wenniger**
Aligning the Foundations of Hierarchical Statistical Machine Translation

ILLC DS-2016-07: **Andreas van Cranenburgh**
Rich Statistical Parsing and Literary Language

ILLC DS-2016-08: **Florian Speelman**
Position-based Quantum Cryptography and Catalytic Computation

ILLC DS-2016-09: **Teresa Piovesan**
Quantum entanglement: insights via graph parameters and conic optimization

ILLC DS-2016-10: **Paula Henk**
Nonstandard Provability for Peano Arithmetic. A Modal Perspective

ILLC DS-2017-01: **Paolo Galeazzi**
Play Without Regret

ILLC DS-2017-02: **Riccardo Pinosio**
The Logic of Kant's Temporal Continuum

ILLC DS-2017-03: **Matthijs Westera**
Exhaustivity and intonation: a unified theory

ILLC DS-2017-04: **Giovanni Cinà**
Categories for the working modal logician

ILLC DS-2017-05: **Shane Noah Steinert-Threlkeld**
Communication and Computation: New Questions About Compositionality

ILLC DS-2017-06: **Peter Hawke**
The Problem of Epistemic Relevance

ILLC DS-2017-07: **Aybüke Özgün**
Evidence in Epistemic Logic: A Topological Perspective

ILLC DS-2017-08: **Raquel Garrido Alhama**
Computational Modelling of Artificial Language Learning: Retention, Recognition & Recurrence

ILLC DS-2017-09: **Miloš Stanojević**
Permutation Forests for Modeling Word Order in Machine Translation

ILLC DS-2018-01: **Berit Janssen**
Retained or Lost in Transmission? Analyzing and Predicting Stability in Dutch Folk Songs

ILLC DS-2018-02: **Hugo Huurdeman**
Supporting the Complex Dynamics of the Information Seeking Process

ILLC DS-2018-03: **Corina Koolen**
Reading beyond the female: The relationship between perception of author gender and literary quality

ILLC DS-2018-04: **Jelle Bruineberg**
Anticipating Affordances: Intentionality in self-organizing brain-body-environment systems

ILLC DS-2018-05: **Joachim Daiber**
Typologically Robust Statistical Machine Translation: Understanding and Exploiting Differences and Similarities Between Languages in Machine Translation

ILLC DS-2018-06: **Thomas Brochhagen**
Signaling under Uncertainty

ILLC DS-2018-07: **Julian Schlöder**
Assertion and Rejection

ILLC DS-2018-08: **Srinivasan Arunachalam**
Quantum Algorithms and Learning Theory