Quantum Information Processing (2020) 19:172
https://doi.org/10.1007/s11128-020-02661-1

Quantum learning Boolean linear functions w.r.t. product distributions

Matthias C. Caro[1]

Received: 5 August 2019 / Accepted: 29 March 2020 / Published online: 20 April 2020

Abstract

The problem of learning Boolean linear functions from quantum examples w.r.t. the uniform distribution can be solved on a quantum computer using the Bernstein–Vazirani algorithm (Bernstein and Vazirani, in: Kosaraju (ed) Proceedings of the twenty-fifth annual ACM symposium on theory of computing, ACM, New York, 1993. https://doi.org/10.1145/167088.167097). A similar strategy can be applied in the case of noisy quantum training data, as was observed in Grilo et al. (Learning with errors is easy with quantum samples, 2017). However, extensions of these learning algorithms beyond the uniform distribution have not yet been studied. We employ the biased quantum Fourier transform introduced in Kanade et al. (Learning dnfs under product distributions via μ-biased quantum Fourier sampling, 2018) to develop efficient quantum algorithms for learning Boolean linear functions on n bits from quantum examples w.r.t. a biased product distribution. Our first procedure is applicable to any (except full) bias and requires $\mathcal{O}(\ln(n))$ quantum examples. The number of quantum examples used by our second algorithm is independent of n, but the strategy is applicable only for small bias. Moreover, we show that the second procedure is stable w.r.t. noisy training data and w.r.t. faulty quantum gates. This also enables us to solve a version of the learning problem in which the underlying distribution is not known in advance. Finally, we prove lower bounds on the classical and quantum sample complexities of the learning problem. Whereas classically, $\Omega(n)$ examples are necessary independently of the bias, we are able to establish a quantum sample complexity lower bound of $\Omega(\ln(n))$ only under an assumption of large bias. Nevertheless, this allows for a discussion of the performance of our suggested learning algorithms w.r.t. sample complexity. With our analysis, we contribute to a more quantitative understanding of the power and limitations of quantum training data for learning classical functions.

Keywords Computational learning theory · Exact learning · Quantum Fourier learning

✉ Matthias C. Caro
caro@ma.tum.de

Extended author information available on the last page of the article

1 Introduction

The origins of the fields of machine learning as well as quantum information and computation both lie in the 1980s. The arguably most influential learning model, namely the PAC ("probably approximately correct") model, was introduced by Valiant in 1984 [26] with which the problem of learning was given a rigorous mathematical framework. Around the same time, Benioff [7] and Feynman presented the idea of quantum computers [12] to the public and thus gave the starting signal for important innovations at the intersection of computer science, information theory and quantum theory. Both learning theory and quantum computation promise new realms of computation in which tasks that seem insurmountable from the perspective of classical computation become feasible. The first has already proved its practical worth and is indispensable for modern-world big data applications, the latter is not yet as practically relevant but much work is invested to make the promises of quantum computation a reality. The interested reader is referred to [20,25] for an introduction to statistical learning and quantum computation and information, respectively.

Considering the increasing importance of machine learning and quantum computation, attempting a merger of the two seems a natural step to take and the first step in this direction was taken already in [10]. The field of quantum learning has received growing attention over the last few years and by now some settings are known in which quantum training data and the ability to perform quantum computation can be advantageous for learning problems from an information-theoretic as well as from a computational perspective, in particular for learning problems with fixed underlying distribution (see, e.g., [3] for an overview). It was, however, shown in [4] that no such information-theoretic advantage can be obtained in the (distribution-independent) quantum PAC model (based on [10]) compared to the classical PAC model (introduced in [26]).

One of the early examples of the aptness of quantum computation for learning problems is the task of learning Boolean linear functions w.r.t. the uniform distribution via the Bernstein–Vazirani algorithm presented in [8]. Whereas this task of identifying an unknown n-bit string classically requires a number of examples growing (at least) linearly with n, a bound on the sufficient number of copies of the quantum example state independent of n can be established. This approach was taken up in [13] where it is shown that, essentially, the Bernstein–Vazirani-based learning method is also viable if the training data is noisy. However, also this analysis is restricted to quantum training data arising from the uniform distribution. The same limiting assumption was also made in [10] for learning Disjunctive Normal Forms and in this context an extension to product distributions was achieved in [17].

Hence, a next direction to go is building up on the reasoning of [17] to extend the applicability of quantum learning procedures for linear functions to more general distributions. The analysis hereby differs from the one for DNFs because no concentration results for the biased Fourier spectrum of a linear function are available. Moreover, whereas many studies of specific quantum learning tasks focus on providing explicit learning procedures yielding a better performance than known classical algorithms, we complement our learning algorithms with lower bounds on the size of the training

data for a comparison to the best classical procedure and for a discussion of optimality among possible quantum strategies.

1.1 Overview over the results

The task of learning linear functions has already served as a toy model for quantum speed-ups in the early days of quantum computing. We describe possible generalizations of known results in different scenarios. First, in Theorem 3 we exhibit a Fourier-sampling-based algorithm which learns Boolean linear functions on n inputs from $\mathcal{O}(\ln(n))$ quantum examples arising from a c-bounded product distribution D_μ. (Classically, it is known that $\Omega(n)$ examples are required.) Moreover, for a bias vector μ satisfying $|\mu_i| \leq \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ for all i, this can be reduced to $\mathcal{O}(1)$ quantum examples (Theorem 4). We also show that this reduction to a constant number of quantum examples is not possible for arbitrary product distributions by giving quantum sample complexity lower bounds in Theorem 6.

In Theorem 8, we exhibit a noise bound for quantum examples arising from a product distribution D_μ with $|\mu_i| \leq \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ for all i but corrupted by noise which guarantees that $\mathcal{O}(1)$ quantum examples still suffice for learning. Under milder assumptions on the noise, a $\mathcal{O}(\ln(n))$ upper bound on the sample complexity is given. Similarly, faulty quantum gates can be tolerated in our learning algorithm. Based on this observation, we construct a quantum learning algorithm without prior knowledge of the underlying distribution which requires $\mathcal{O}(n^2)$ quantum examples by first estimating the bias vector classically (Corollary 3).

1.2 Related work

The (classical) problem of learning linear functions from randomly drawn examples in the presence of noise was studied in [9] (over the field $\mathbb{F}_2$) as well as in [22] (over a field $\mathbb{F}_q$ for q prime). The latter of these two works also established the relevance of this learning problem for cryptography by connecting it to certain lattice problems. A different model for learning linear functions is studied in [16], where the training data is not assumed to be noisy but instead only partial information about the function values is revealed.

The quantum PAC model was introduced in [10], where it was employed for learning DNF formulae w.r.t. the uniform distribution using a quantum example oracle. This was extended to product distributions by [17]. On the basis of this notion of quantum examples, the known Bernstein–Vazirani algorithm [8] can be reinterpreted as giving rise to a quantum learning algorithm for linear functions. This interpretation is explicitly given and further elaborated upon for the case of noisy training data in [11] (for $q = 2$) and in [13] (for general primes q). Cross et al. [11] established that, whereas the learning parity problem without noise is feasible both for classical and quantum computation, the learning parity with noise problem is widely believed to be classically intractable but remains feasible for quantum computers, where the runtime depends only logarithmically on the number of qubits. This quantum advantage for

noisy systems was demonstrated experimentally in [23]. Grilo et al. [13] extends this analysis to general fields and a broader class of noise models and obtains that also for that scenario, learning linear functions from noisy data is feasible for quantum computers; however, their runtime bound is polynomial in the number of subsystems. In [5], the class of juntas is found to also allow for efficient quantum learning. The framework of Fourier-based quantum exact learning is shown to be efficiently applicable more generally also to Fourier-sparse functions in [1]. Limitations of the power of quantum computation for learning have been studied in a series of papers culminating in [4] and more recently also in [2]. The former work shows that without prior restrictions on the underlying probability distribution, quantum examples are not more powerful than classical examples. The latter work demonstrates that, assuming quantum hardness of the learning with errors problem from classical examples, the class of shallow circuits is hard to learn from quantum examples.

Aside from the task of learning from examples, also the problem of learning from membership queries, both classical and quantum, is well studied. For instance, [24] established a polynomial relation between the number of required quantum versus required classical queries, which was recently improved upon in [1]. Also, [19] uses quantum membership queries for learning multilinear polynomials more efficiently than is classically possible.

1.3 Structure of the paper

The paper is structured in the following way. In Sect. 2, we introduce the well-known notions from classical learning, quantum computation and Boolean Fourier analysis required for our purposes as well as the prototypic learning algorithm which motivates our procedures. Section 3 consists of a description of the learning task to be considered. This is followed by a generalization of the Bernstein–Vazirani algorithm to product distributions in Sect. 4. In the next section, this is used to develop two quantum algorithms for solving our problem. ("Appendix A" contains a stability analysis of the second of the two procedures w.r.t. noise in training data and computation.) In Sect. 6, we establish sample complexity lower bounds complementing the upper bounds implied by the algorithms of Sect. 5. Finally, we conclude with some open questions and the references.

2 Preliminaries

2.1 Basics of quantum information and computation

We first define some of the fundamental objects of quantum information theory, albeit restricted to those required in our discussion. For the purpose of our presentation, we will consider a pure n-qubit quantum state to be represented by a state vector $|\psi\rangle \in \mathbb{C}^{2^n}$ (in Dirac notation). Such a state encodes measurement probabilities in the following way: If $\{|b_i\rangle\}_{i=1}^{2^n}$ is an orthonormal basis of $\mathbb{C}^{2^n}$, then there corresponds a measurement to this basis and the probability of observing outcome i for a system in

state $|\psi\rangle$ is given by $|\langle b_i|\psi\rangle|^2$. Finally, when considering multiple subsystems we will denote the composite state by the tensor product, i.e., if the first system is in state $|\psi\rangle$ and the second in state $|\phi\rangle$, the composite system is in state $|\psi, \varphi\rangle := |\psi\rangle \otimes |\phi\rangle$.

Quantum computation now consists in evolution of quantum states. Performing a computational step on an n-qubit state corresponds to applying an $2^n \times 2^n$ unitary transformation to the current quantum state. (The most relevant example of such unitary gates in our context will be the (biased) quantum Fourier transform discussed in more detail in Sect. 2.4.) As the outcome of a quantum computation is supposed to be classical, as final step of our computation we perform a measurement such that the final output will be a sample from the corresponding measurement statistics.

We will also use some standard notions from (quantum) information theory. For example, we denote the Shannon entropy of a random variable X by $H(X)$, the conditional entropy of a random variable X given Y as $H(X|Y)$ and the mutual information between random variables X and Y as $I(X : Y)$. Similarly, the von Neumann entropy of a quantum state ρ will be denoted as $S(\rho)$ and the mutual information for a bipartite quantum state ρ_{AB} as $I(\rho_{AB}) = I(A : B)$. Standard results on these quantities which will enter our discussion can, e.g., be found in [20].

2.2 Basics of learning theory

Next we describe the model of exact learning. In classical exact learning for an input space $\mathcal{X}$, a target space $\{0, 1\}$, and a concept class $\mathcal{F} \subset \{0, 1\}^{\mathcal{X}}$, a learning algorithm receives as input labeled training data $\{(x_i, f(x_i))\}_{i=1}^m$ for some (to the learner) unknown $f \in \mathcal{F}$, where the x_i are drawn independently according to some probability distribution D on $\mathcal{X}$ which is known to the learner. The goal of the learner is to exactly reproduce the unknown function f from such training examples with high success probability.

We can formalize this as follows: We call a concept class $\mathcal{F}$ exactly learnable if there exists a learning algorithm $\mathcal{A}$ and a map $m_{\mathcal{F}} : (0, 1) \to \mathbb{N}$ s.t. for every $D \in \text{Prob}(X)$ (where $\text{Prob}(X)$ is the set of all probability measures on X), $f \in \mathcal{F}$ and $\delta \in (0, 1)$, running $\mathcal{A}$ on training data of size $m \geq m_{\mathcal{F}}(\delta)$ drawn according to D and f with probability $\geq 1 - \delta$ (w.r.t. the choice of training data) yields a hypothesis h s.t. $h(x) = f(x)$ for all $x \in \mathcal{X}$. The smallest such map $m_{\mathcal{F}}$ is called sample complexity of exactly learning $\mathcal{F}$.

Note that this definition of learning captures the information-theoretic challenge of the learning problem in the sample complexity, but it does not refer to the computational complexity of learning. The focus on sample complexity is typical in statistical learning theory. Hence, also our results will be formulated in terms of sample complexity bounds. As we give explicit algorithms, these results directly imply bounds on the computational complexity; however, we will not discuss them in any detail.

Note also that the exact learning model differs from the well-known PAC ("probably approximately correct"), introduced by [26], in two ways. First, whereas the PAC model only requires to approximate the unknown function with high probability, we require to reproduce it exactly; in other words, we set the accuracy in PAC learning

to 0. Second, whereas in the PAC scenario the learner does not know the underlying distribution, we assume it to be fixed and known in advance. A short discussion on how to relax this restriction can be found in Sect. A.3.

The quantum exact learning model differs from the classical model in the form of the training data and the allowed form of computation. Namely, in quantum exact learning, the training data consists of m copies of the quantum example state $|\psi_f\rangle = \sum_{x\in\mathcal{X}} \sqrt{D(x)}|x, f(x)\rangle$, and this training data is processed by quantum computational steps. With this small change, the above definition of exact learnability and sample complexity now carry over analogously.

We conclude this introduction with a concentration result that has proven to be useful throughout learning theory.

Lemma 1 (Hoeffding's Inequality [15], compare also Theorem 2.2.6 in [27])
Let $Z_1, ..., Z_n$ be real-valued independent random variables taking values in closed and bounded intervals $[a_i, b_i]$, respectively. Then for every $\varepsilon > 0$

$$\mathbb{P}\left[\sum_{i=1}^{n} Z_i - \mathbb{E}[Z_i] \geq \varepsilon\right] \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{n}(a_i - b_i)^2}\right).$$

This directly implies (after replacing Z_i with $-Z_i$) that

$$\mathbb{P}\left[\left|\sum_{i=1}^{n} Z_i - \mathbb{E}[Z_i]\right| \geq \varepsilon\right] \leq 2\exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{n}(a_i - b_i)^2}\right).$$

2.3 μ-biased Fourier analysis of Boolean functions

We now give the basic ingredients of μ-biased Fourier analysis over the Boolean cube $\{-1, 1\}^n$. For more details, the reader is referred to [21].
For a bias vector $\mu \in [-1, 1]^n$, define the μ-biased product distribution D_μ on $\{-1, 1\}^n$ via

$$D_\mu(x) := \left(\prod_{i:x_i=1} \frac{1+\mu_i}{2}\right)\left(\prod_{i:x_i=-1} \frac{1-\mu_i}{2}\right) = \prod_{1\leq i\leq n} \frac{1+x_i\mu_i}{2}, \quad x \in \{-1, 1\}^n.$$

Thus, a positive μ_i tells us that at the ith position the distribution is biased towards $+1$, a negative μ_i tells us that at the ith position the distribution is biased towards -1. For $\mu = 0 \ldots 0$, we simply obtain the uniform distribution on $\{-1, 1\}^n$. The absolute value of μ_i quantifies the strength of the bias in the ith component. We call D_μ c-bounded, for $c \in (0, 1]$, if $\mu \in [-1 + c, 1 - c]^n$. Assuming the underlying product distribution to be c-bounded thus corresponds to assuming that the bias is not arbitrarily strong. Hence, we will in the following express notions of "small" or "large" bias either in terms of the bias vector μ or in terms of the c-boundedness constant.

For Fourier analysis, we now need an orthonormal basis for the function space $\mathbb{R}^{\{-1,1\}^n}$ w.r.t. the inner product $\langle .,. \rangle_\mu$ defined by

$$\langle f, g \rangle_\mu = \mathbb{E}_{D_\mu}[fg] = \sum_{x \in \{-1,1\}^n} f(x)g(x)D_\mu(x).$$

One can show (using the product structure to reduce to the case $n = 1$) that such an orthonormal basis is given by $\{\phi_{\mu,j}\}_{j \in \{0,1\}^n}$ with $\phi_{\mu,j}(x) = \prod_{i:j_i=1} \frac{x_i - \mu_i}{\sqrt{1-\mu_i^2}}$. For a function $f : \{-1,1\}^n \to \{-1,1\}$ this now gives a representation $f(x) = \sum_{j \in \{0,1\}^n} \hat{f}_\mu(j)\phi_{\mu,j}(x)$ with $\hat{f}_\mu(j) := \langle f, \phi_{\mu,j} \rangle_\mu$. For $\mu = 0 \ldots 0$, we recover the well-known orthonormal basis consisting of $\chi_j(x) = (-1)^{j \cdot x}$ from standard Fourier analysis over the Boolean cube.

2.4 μ-biased quantum Fourier sampling

We now turn to the description of the quantum algorithm for μ-biased quantum Fourier sampling which constitutes the basic ingredient of our learning algorithms and which, to our knowledge, was first presented in [17]. There the authors demonstrate that the μ-biased Fourier transform for a c-bounded D_μ with $c \in (0, 1]$ can be implemented on a quantum computer as the n-qubit μ-biased quantum Fourier transform: For $x \in \{-1, 1\}^n$,

$$H_\mu^n |x\rangle = H_\mu \otimes \ldots \otimes H_\mu |x_1, \ldots, x_n\rangle = \sum_{j \in \{0,1\}^n} \sqrt{D_\mu(x)}\phi_{\mu,j}(x)|j\rangle.$$

In the same way as the unbiased quantum Fourier transform can be used for quantum Fourier sampling, this μ-biased version now yields a procedure to sample from the μ-biased Fourier spectrum of a function using a quantum computer. We describe the corresponding procedure in Algorithm 1.

Algorithm 1 μ-biased Quantum Fourier Sampling

Input: $|\psi_f\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)}|x, f(x)\rangle$ for a function $f : \{-1, 1\}^n \to \{0, 1\}$

Output: $j \in \{0, 1\}^n$ with probability $\left(\hat{g}_\mu(j)\right)^2$, where the function $g : \{-1, 1\}^n \to \{-1, 1\}$ is defined as $g(x) = (-1)^{f(x)}$.

Success Probability: $\frac{1}{2}$

1: Perform the μ-biased QFT H_μ on the first n qubits, obtain the state $(H_\mu \otimes \mathbb{1})|\psi_f\rangle$.
2: Perform a Hadamard gate on the last qubit, obtain the state $(H_\mu \otimes H)|\psi_f\rangle$.
3: Measure each qubit in the computational basis and observe outcome $j = j_1 \ldots j_{n+1}$.
4: **if** $j_{n+1} = 0$ **then** ▷ This corresponds to a failure of the sampling algorithm.
5: Output $o \leftarrow \perp$ and end computation.
6: **else if** $j_{n+1} = 1$ **then** ▷ This corresponds to a success of the sampling algorithm.
7: Output $o \leftarrow j_1 \ldots j_n$ and end computation.
8: **end if**

One can show that this algorithm indeed works as claimed by analyzing the transformation of the quantum state throughout the steps algorithm and making use of the orthonormality of the basis. This is the content of the following

Lemma 2 (Lemma 3 in [17])

Denote $g : \{-1, 1\}^n \to \{-1, 1\}$, $g(x) = (-1)^{f(x)}$. *Then with probability* $\frac{(\hat{g}_\mu(j))^2}{2}$, *Algorithm 1 outputs the string* $j \in \{0, 1\}^n$.

Proof The proof can be found in [17], we reproducce it in "Appendix B." □

This result allows us to generalize results based on quantum Fourier sampling w.r.t. the uniform distribution. In particular, we will apply it to obtain a generalization of the Bernstein–Vazirani algorithm.

2.5 The pretty good measurement

A basic problem in quantum information is that of distinguishing quantum states. We now describe a useful tool in this context, namely a measurement that is guaranteed to have a "pretty good" success probability to correctly identify an unknown state from a known ensemble.

Suppose that Alice (A) chooses one among m pure states $|\psi_i\rangle \in \mathbb{C}^d$ according to probabilities $p_i \in [0, 1]$, where $p_i \geq 0$ and $\sum_{i=1}^m p_i = 1$ and then sends the state to Bob (B). B wants to identify the state by performing a POVM measurement $\mathcal{A}$. Let $\mathcal{E} = \{(p_i, |\psi_i\rangle)\}_{i=1,\ldots,m}$ be the ensemble describing A's preparation procedure, denote B's optimal success probability by $P^{opt} := \max_{POVM\ \mathcal{A}} P^{\mathcal{A}}$, where $P^{\mathcal{A}} := \sum_{i=1}^m p_i \langle\psi_i|A_i|\psi_i\rangle$ for a POVM $\mathcal{A} = \{A_i\}_{i=1,\ldots,m}$. Hausladen and Wootters [14] suggested a canonical form for a measurement for state discrimination, which is now usually referred to as the "pretty good measurement" (PGM) corresponding to the ensemble $\mathcal{E}$. It is defined in the following way:

First let $|\psi_i'\rangle := \sqrt{p_i}|\psi_i\rangle$ be the states renormalized according to their respective probabilities. The density operator of the ensemble $\mathcal{E}$ is $\rho := \sum_{i=1}^m p_i |\psi_i\rangle\langle\psi_i| = \sum_{i=1}^m |\psi_i'\rangle\langle\psi_i'|$. Now define $|\varphi_i\rangle := \rho^{-\frac{1}{2}}|\psi_i\rangle$, where the inverse square root is taken only over nonzero eigenvalues of ρ. Now the PGM is $\mathcal{A}^{PGM} = \{|\varphi_i\rangle\langle\varphi_i|\}_{i=1,\ldots,m}$. (Observe that this is indeed a valid POVM, even a projection-valued measure (PVM), because $\sum_{i=1}^m |\varphi_i\rangle\langle\varphi_i| = \rho^{-\frac{1}{2}}\rho\rho^{-\frac{1}{2}} = \mathbb{1}_d$.)

The "pretty good" performance of the PGM was proved in [6]:

Theorem 1 *For the PGM measurement defined above it holds that*

$$P^{opt}(\mathcal{E})^2 \leq P^{PGM}(\mathcal{E}) \leq P^{opt}(\mathcal{E}).$$

Another useful property of the PGM is that the corresponding success probability can be computed from the Gram matrix of the ensemble as follows:

Lemma 3 *The success probability for the PGM measurement for an ensemble* $\mathcal{E} = \{(p_i, |\psi_i\rangle)\}_{i=1\ldots,m}$ *can be written as*

$$P^{PGM}(\mathcal{E}) = \sum_{i=1}^{m} \sqrt{G}(i,i)^2,$$

where G *is the Gram matrix with entries* $G(i,j) = \sqrt{p_i p_j}\langle\psi_i|\psi_j\rangle$ *for* $1 \leq i, j \leq m$.

Proof This result can be shown by direct computation using the definition of the PGM and the uniqueness of the positive square root of a positive matrix. □

3 The learning problem

We now describe the learning task which we aim to understand. For $a \in \{0,1\}^n$, define

$$f^{(a)} : \{-1,1\}^n \to \{0,1\}, \; f^{(a)}(x) := \sum_{i=1}^{n} a_i \frac{1-x_i}{2} \pmod 2.$$

When we observe that $\frac{1-x_i}{2}$ is simply the bit-description of x_i, it becomes clear that $f^{(a)}$ computes the parity of the entries of the bit-description of x_i at the positions at which a has a 1-entry. To ease readability, we will write $\tilde{x}_i = \frac{1-x_i}{2}$.

The classical task which inspires our problem is the following: Given a set of m labeled examples $S = \{(x_i, f^{(a)}(x_i))\}_{i=1}^{m}$, where the x_i are drawn i.i.d. according to D_μ, determine the string a with high success probability. Here, we assume prior knowledge of the underlying distribution and that the underlying distribution is a c-bounded product distribution as introduced in Sect. 2.4. This means that we are considering a problem of exact learning from examples with instances drawn from a distribution that is known to the learner in advance.

Classically, as we show in Sect. 6, successfully solving the task requires a number of examples that grows at least linearly in n. If we consider a version of this problem with noisy training data, then known classical algorithms perform worse both w.r.t. sample complexity and running time. For example, [18] exhibits an algorithm with polynomial (superlinear) sample complexity but barely subexponential runtime (both w.r.t. n).

The step to the quantum version of this problem now is the same as from classical to quantum exact learning. This means that training data is given as m copies of the quantum example state $|\psi_a\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, f^{(a)}(x)\rangle$ and the learner is allowed to use quantum computation to process the training data. The goal of the quantum learner remains that of outputting the unknown string a with high success probability.

4 A generalized Bernstein–Vazirani algorithm

To understand how μ-biased quantum Fourier sampling can help us with this learning problem, we first compute the μ-biased Fourier coefficients of $g^{(a)} := (-1)^{f^{(a)}}$, with $f^{(a)}$ for $a \in \{0,1\}^n$ the linear functions defined in Sect. 3.

Lemma 4 *Let $a \in \{0,1\}^n$, $g^{(a)} := (-1)^{f^{(a)}}$ and $\mu \in (-1,1)^n$. Then the μ-biased Fourier coefficients of $g^{(a)}$ satisfy:*

(i) *If $\exists\, 1 \le i \le n$ s.t. $a_i = 0 \neq j_i$, then $\hat{g}_\mu^{(a)}(j) = 0$.*
(ii) *If for all $1 \le i \le n$ s.t. $a_i = 0$ also $j_i = 0$, then*

$$\hat{g}_\mu^{(a)}(j) = \left(\prod_{l:a_l=1\neq j_l} \mu_l\right)\left(\prod_{l:a_l=1=j_l} \sqrt{1-\mu_l^2}\right).$$

We can reformulate this as

$$\hat{g}_\mu^{(a)}(j) = \left(\prod_{l:a_l=0} (1-j_l)\right)\left(\prod_{l:a_l=1} \left((1-j_l)\mu_l + j_l\sqrt{1-\mu_l^2}\right)\right), \quad j \in \{0,1\}^n.$$

Proof We first observe that all the "objects of interest," namely the probability distribution D_μ, the basis functions $\phi_{\mu,j}$, and the target function $\hat{g}_\mu^{(a)}$, factorize. This now implies that also the μ-biased Fourier coefficients factorize, i.e., we have

$$\hat{g}_\mu^{(a_1\ldots a_n)}(j_1\ldots j_n) = \prod_{i=1}^{n} \mathbb{E}_{D_{\mu_i}}[\phi_{\mu_i,j_i}(x_i)\cdot(-1)^{a_i\cdot\tilde{x}_i}].$$

Therefore we only have to study the case $n = 1$ in detail and the general result then follows. In this case, we have $f^{(a)}(x) = a\tilde{x}$, $g^{(a)}(x) = (-1)^{a\tilde{x}}$ for $\tilde{x} = \frac{1-x}{2}$, $\phi_{\mu,0}(x) = 1$, and $\phi_{\mu,1}(x) = \frac{x-\mu}{\sqrt{1-\mu^2}}$. (We leave out unnecessary indices to improve readability.) We compute

$$\hat{g}_\mu^{(a)}(j) = \mathbb{E}_{D_\mu}[(-1)^{a\tilde{x}}\phi_{\mu,j}(x)] = \frac{1+\mu}{2}\cdot 1\cdot\phi_{\mu,j}(1) + \frac{1-\mu}{2}\cdot(-1)^a\cdot\phi_{\mu,j}(-1).$$

By plugging in we now obtain

$$\hat{g}_\mu^{(0)}(0) = 1, \quad \hat{g}_\mu^{(0)}(1) = 0, \quad \hat{g}_\mu^{(1)}(0) = \mu, \quad \hat{g}_\mu^{(1)}(1) = \sqrt{1-\mu^2},$$

which is exactly the claim for $n = 1$. □

For clarity, we write down explicitly the algorithm which we obtain as a generalization of the Bernstein–Vazirani algorithm to a μ-biased product distribution as

Algorithm 2. The generalization compared to the standard Bernstein–Vazirani algorithm consists only in going from the uniform to a more general product distribution, which gives rise to different observation probabilities.

Algorithm 2 Generalized Bernstein–Vazirani algorithm

Input: $|\psi_a\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, f^{(a)}(x)\rangle$ for $a \in \{0,1\}^n$, and $\mu \in [-1,1]^n$

Output: $o \in \{0,1\}^n$ with probability

$$\left(\prod_{l:a_l=0}(1-o_l)\right)\left(\prod_{l:a_l=1}\left((1-o_l)\mu_l^2 + o_l(1-\mu_l^2)\right)\right)$$

Success Probability: $\frac{1}{2}$

1: Perform the μ-biased QFT H_μ on the first n qubits, obtain the state $(H_\mu \otimes \mathbb{1})|\psi_a\rangle$.
2: Perform a Hadamard gate on the last qubit, obtain the state $(H_\mu \otimes H)|\psi_a\rangle$.
3: Measure each qubit in the computational basis and observe outcome $j = j_1 \dots j_{n+1}$.
4: **if** $j_{n+1} = 0$ **then** ▷ This corresponds to a failure of the algorithm.
5: Output $o = \perp$.
6: **else if** $j_{n+1} = 1$ **then** ▷ This corresponds to a success of the algorithm.
7: Output $o = j_1 \dots j_n$.
8: **end if**

We now show that the output probabilities of Algorithm 2 are as claimed in its description. This follows directly by combining Lemma 2 on the workings of μ-biased quantum Fourier sampling with Lemma 4 on the μ-biased Fourier coefficients of our target functions and is the content of the following

Theorem 2 *Let* $|\psi_a\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, f^{(a)}(x)\rangle$ *be a quantum example state, with* $a \in \{0,1\}^n$ *and* $\mu \in (-1,1)^n$. *Then step* 3 *of Algorithm* 2 *provides an outcome* $|j_1 \dots j_{n+1}\rangle$ *with the following properties:*

(i) $\mathbb{P}[j_{n+1} = 0] = \frac{1}{2} = \mathbb{P}[j_{n+1} = 1]$,

(ii) $\mathbb{P}[j_1 \dots j_n = a | j_{n+1} = 1] = \prod_{l:a_l=1}(1-\mu_l^2)$,

(iii) *for* $o \neq a$:

$$\mathbb{P}[j_1 \dots j_n = o | j_{n+1} = 1] = \prod_{l:a_l=0}(1-o_l) \cdot \prod_{l:a_l=1}\left((1-o_l)\mu_l^2 + o_l(1-\mu_l^2)\right),$$

(iv) $\mathbb{P}[\exists 1 \le i \le n : a_i = 0 \neq j_i | j_{n+1} = 1] = 0$, *and*

(v) $\mathbb{P}[\exists 1 \le i \le n : a_i = 1 \neq j_i | j_{n+1} = 1] \le \sum_{i=1}^n \mu_i^2$. *In particular, if* D_μ *is* c*-bounded, then* $\mathbb{P}[\exists 1 \le i \le n : a_i = 1 \neq j_i | j_{n+1} = 1] \le n(1-c)^2$.

Note that (v) can be trivial if the bias is too strong. This observation already hints at why we later use different procedures for arbitrary and for small bias.

We also want to point out that in the case of no bias (i.e., $\mu = 0$), Algorithm 2 simply reduces to the well-known Bernstein–Vazirani algorithm [8].

5 Quantum sample complexity upper bounds

This section contains the description of two procedures for solving the task of learning an unknown Boolean linear function from quantum examples w.r.t. a product distribution. (Here, we assume perfect quantum examples, noisy examples will be taken into consideration in the next section.) It is subdivided into an approach which is applicable for arbitrary (albeit not full) bias in the product distribution and a strategy which produces better results but is only valid for small bias.

5.1 Arbitrary bias

As in the case of learning w.r.t. the uniform distribution, we intend to run the generalized Bernstein–Vazirani algorithm multiple times as a subroutine and then use our knowledge of the outcome of the subroutine together with probability-theoretic arguments. The main difficulty compared to the case of an example state arising from the uniform distribution lies in the fact that whereas an observation of $j_{n+1} = 1$ when performing the standard Bernstein–Vazirani algorithm guarantees that $j_1 \dots j_n$ equals the desired string, this is not true in the μ-biased case. Hence, we have to develop a different procedure of learning from the outcomes of the subroutine. For this purpose, we propose Algorithm 3.

Algorithm 3 Amplified Generalized Bernstein–Vazirani algorithm - Version 1

Input: m copies of $|\psi_a\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, f^{(a)}(x)\rangle$ for $a \in \{0,1\}^n$, where the number of copies is $m \geq C\left(\left\lceil\left(2\ln\left(\frac{1}{1-c+\frac{c^2}{2}}\right)\right)^{-1}\left(\ln(n)+\ln(\frac{2}{\delta})\right)\right\rceil\right)$ for a suitable constant $C > 0$, and $\mu \in (-1,1)^n$ and $c \in (0,1]$ s.t. D_μ is c-bounded.
Output: $a \in \{0,1\}^n$
Success Probability: $\geq 1-\delta$

1: **for** $1 \leq l \leq m$ **do**
2: Run Algorithm 2 on the lth copy of $|\psi_a\rangle$, store the output as $o^{(l)}$.
3: **end for**
4: **if** $\exists 1 \leq l \leq m : o^{(l)} \neq \perp$ **then**
5: **for** $1 \leq i \leq n$ **do**
6: Let $o_i := \max_{l:o^{(l)}\neq\perp} o_i^{(l)}$.
7: **end for**
8: Output $o = o_1 \dots o_n$.
9: **else if** $\forall 1 \leq l \leq m : o^{(l)} = \perp$ **then**
10: Output $o = \perp$.
11: **end if**

The amplification procedure in Algorithm 3 differs from the majority vote in the standard Bernstein–Vazirani learning procedure (w.r.t. the uniform distribution) as used in [11,13] in the following two ways: Instead of working on the level of the whole string, we use a componentwise strategy. And instead of taking a majority

vote over observed values, we take a maximum to account for the asymmetry in the probability of an observation error (see Theorem 2).

We now show that the number of copies postulated in Algorithm 3 is actually sufficient to achieve the desired success probability.

Theorem 3 *Let* $|\psi_a\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, f^{(a)}(x)\rangle$, $a \in \{0,1\}^n$, $\mu \in (-1,1)^n$ *s.t.* D_μ *is c-bounded for some* $c \in (0,1]$. *Then*

$$\mathcal{O}\left(\left(2\ln\left(\frac{1}{1-c+\frac{c^2}{2}}\right)\right)^{-1}\left(\ln(n)+\ln(\frac{2}{\delta})\right)\right)$$

copies of the quantum example state $|\psi_a\rangle$ *are sufficient to guarantee that, with probability* $\geq 1-\delta$, *Algorithm 3 outputs the string a.*

Proof We want to show that $\mathbb{P}[\text{Algorithm 3 does not output } a] \leq \delta$. We do so by treating separately the cases in which Algorithm 3 does not output a.
The first such case occurs if $o = \perp$. The second such case would be that there exists $1 \leq i \leq n$ s.t. $a_i = 0 \neq o_i$, but due to Theorem 2, this is an event of probability 0. The third and last such case is that there exists $1 \leq i \leq n$ s.t. $a_i = 1 \neq o_i$. Hence, we can decompose the probability of Algorithm 3 producing a wrong output as

$$\begin{aligned}&\mathbb{P}[\text{Algorithm 3 does not output } a]\\ &= \mathbb{P}[\text{Algorithm 3 outputs } \perp] + \mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq o_i].\end{aligned} \tag{5.1}$$

First, we bound the probability of the algorithm outputting $\perp$ (i.e., of each subroutine failing) as follows:

$$\begin{aligned}&\mathbb{P}[\text{Algorithm 3 outputs } \perp]\\ &= \mathbb{P}[\forall 1 \leq l \leq m : \text{Algorithm 2 applied to } |\psi_a\rangle \text{ outputs } \perp]\\ &= \left(\frac{1}{2}\right)^m,\end{aligned}$$

where the last step uses Theorem 2 and that the training data consists of independent copies of $|\psi_a\rangle$, i.e., is given as a product state. The choice of m now guarantees that this last term is $\leq \frac{\delta}{2}$ (if we choose the constant $C > 0$ sufficiently large).

Now we bound the second term in Eq. (5.1). We make the following observation: Suppose $1 \leq i \leq n$ is s.t. $a_i = 1$. As the Fourier coefficients, and with them the output probabilities, factorize, the probability of Algorithm 2 outputting a string $j_1 \ldots j_n$ with $j_i = 1 = a_i$ is simply the probability of Algorithm 2 applied to only the subsystem state of $|\psi_a\rangle$ corresponding to the ith and the $(n+1)^{st}$ subsystem outputting a 1. By Theorem 2, this probability is

$$\mathbb{P}[j_i = 1] = \mathbb{P}[j_{n+1} = 1] \cdot \mathbb{P}[j_i = 1 | j_{n+1} = 1] = \frac{1}{2} \cdot (1 - \mu_i^2).$$

Hence, assuming $a_i = 1$, the probability of not observing a 1 at the ith position in any of the m runs of Algorithm 2 is $\left(1 - \frac{1}{2} \cdot (1 - \mu_i^2)\right)^m = \left(\frac{1}{2}(1 + \mu_i^2)\right)^m$. By c-boundedness of the distribution D_μ we get

$$\left(\frac{1}{2}(1 + \mu_i^2)\right)^m \leq \left(\frac{1}{2} + \frac{1}{2}(1 - c)^2\right)^m = \left(1 - c + \frac{c^2}{2}\right)^m.$$

So using the union bound, we arrive at

$$\begin{aligned}
&\mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq o_i] \\
&\quad = \mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \text{ and in } m \text{ runs no 1 is observed at the } i^{th} \text{ entry}] \\
&\quad \leq \sum_{i=1}^{n} \mathbb{P}[a_i = 1 \text{ and in } m \text{ runs no 1 is observed at the } i^{th} \text{ entry}] \\
&\quad \leq n \cdot \left(1 - c + \frac{c^2}{2}\right)^m.
\end{aligned}$$

The choice of m guarantees that this last term is $\leq \frac{\delta}{2}$ (if we choose the constant $C > 0$ sufficiently large).

We now combine this with Eq. (5.1) and obtain

$$\mathbb{P}[\text{Algorithm 3 does not output } a] \leq \frac{\delta}{2} + \frac{\delta}{2} = \delta,$$

which finishes the proof. □

Remark 1 We want to comment shortly on the dependence of the sample complexity bound on the c-boundedness constant by considering extreme cases. As $c \to 0$, i.e., we allow more and more strongly biased distributions, the sample complexity goes to infinity. This reflects the fact that in the case of a fully biased underlying product distribution, only a single bit of information about a can be extracted, so exactly learning the string a is (in general) not possible.
For $c = 1$, i.e., the case of no bias, we simply obtain that $\mathcal{O}\left(\left(\ln(n) + \ln(\frac{2}{\delta})\right)\right)$ copies of the quantum example state are sufficient. Note that this does not coincide with the bound obtained for the standard Bernstein–Vazirani procedure which is independent of n. (This can easily be shown using Lemma 1.)

This discrepancy is due to the difference in "amplification procedures." Namely, in Algorithm 3 we do not explicitly make use of the knowledge that, given $j_{n+1} = 1$, we know the probability of $j_1 \ldots j_n = a_1 \ldots a_n$ because, whereas for $\mu = 0$ this probability equals 1, for $\mu \neq 0$ it can become small. Hence, for $\mu \neq 0$ our algorithm introduces an additional procedure to deal with the uncertainty of $j_1 \ldots j_n$ even knowing j_{n+1} and we see in the proof that this yields the additional $\ln(n)$ term. In the next subsection, we describe a way to get rid of exactly that $\ln(n)$ term for "small" bias.

5.2 Small bias

In this subsection, we want to study the case in which (v) of Theorem 3 gives a good bound. Namely, throughout this subsection we will assume that the c-boundedness constant is s.t. $n(1-c)^2 < \frac{1}{2}$ or, equivalently, $c > 1 - \frac{1}{\sqrt{2n}}$. This assumption will allow us to apply a different procedure to learn from the output of Algorithm 2 and thus obtain a different bound on the sample complexity of the problem. Note, however, that this requirement becomes more restrictive with growing n and can in the limit $n \to \infty$ only be satisfied by $c = 1$, i.e., for the underlying distributions being uniform. Also, we will from now on refer to c as c-boundedness parameter because the name "constant" would hide the n-dependence.

Our procedure for the case of small bias is given in Algorithm 4.

Algorithm 4 Amplified Generalized Bernstein–Vazirani algorithm - Version 2

Input: m copies of $|\psi_a\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, f^{(a)}(x)\rangle$ for $a \in \{0,1\}^n$, where the number of copies is $m \geq C\left(\frac{4}{(1-2n(1-c)^2)^2} \ln\left(\frac{2}{\delta}\right)\right)$, as well as $\mu \in [-1,1]^n$ and $c \in (0,1]$ s.t. D_μ is c-bounded.
Output: $a \in \{0,1\}^n$
Success Probability: $\geq 1-\delta$

1: **for** $1 \leq l \leq m$ **do**
2: Run Algorithm 2 on the l^{th} copy of $|\psi_a\rangle$, store the output as $o^{(l)}$.
3: **end for**
4: **if** $\exists 1 \leq l \leq m : o^{(l)} \neq \perp$ **then**
5: **for** $1 \leq i \leq n$ **do**
6: Let $o_i = \arg\max_{r\in\{0,1\}} |\{1 \leq l \leq m | o_i^{(l)} = r\}|$.
7: **end for**
8: Output $o = o_1 \ldots o_n$.
9: **else if** $\forall 1 \leq l \leq m : o^{(l)} = \perp$ **then**
10: Output $o = \perp$.
11: **end if**

Theorem 4 *Let* $|\psi_a\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, f^{(a)}(x)\rangle$, $a \in \{0,1\}^n$, $\mu \in (-1,1)^n$ *s.t.* D_μ *is* c*-bounded for some* $c \in (0,1]$ *satisfying* $c > 1 - \frac{1}{\sqrt{2n}}$. *Then*

$$\mathcal{O}\left(\frac{1}{(1-2n(1-c)^2)^2} \ln\left(\frac{1}{\delta}\right)\right)$$

copies of the quantum example state $|\psi_a\rangle$ *are sufficient to guarantee that, with probability* $\geq 1-\delta$, *Algorithm 4 outputs the string* a.

Note that due to the required lower bound on c the sample complexity upper bound basically loses its n-dependence. This is different from the result of Theorem 3, where n explicitly entered the upper bound.

Proof By Theorem 2, we have $\mathbb{P}[j_{n+1} = 1] = \frac{1}{2}$. Hence, the probability of observing $j_{n+1} = 1$ in at most $k-1$ of the m runs of Algorithm 2 is given by

$$\sum_{l=0}^{k-1} \binom{m}{i} \left(\frac{1}{2}\right)^i \left(\frac{1}{2}\right)^{m-i} = \mathbb{P}\left[\mathrm{Bin}(m, \frac{1}{2}) \geq m-k\right],$$

where Bin denotes a binomial distribution.

Next we assume $k \leq \frac{m}{2}$ (this will be justified later in the proof) and use Hoeffding's inequality (Lemma 1) to obtain

$$\begin{aligned}\mathbb{P}\left[\mathrm{Bin}\left(m, \frac{1}{2}\right) \geq m-k\right] &= \mathbb{P}\left[\mathrm{Bin}\left(m, \frac{1}{2}\right) - \frac{m}{2} \geq m-k-\frac{m}{2}\right] \\ &\leq \exp\left(-\frac{2\left(\frac{m}{2}-k\right)^2}{m}\right). \end{aligned} \tag{5.2}$$

We will now search for the number of observations of $j_{n+1} = 1$ which is required to guarantee that the majority string is correct with high probability. Assume that we observe $j_{n+1} = 1$ in k runs of Algorithm 2, $k \in 2\mathbb{N}$. (The latter assumption clearly does not significantly change the number of copies.) Using (v) from Theorem 2, we see that

$$\begin{aligned}\mathbb{P}[\exists 1 \leq i \leq n : a_i \neq o_i] &\leq \mathbb{P}[\exists 1 \leq i \leq n : a_i = 0 \neq o_i] \\ &\quad + \mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq o_i] \\ &\leq 0 + \sum_{l=\lceil \frac{k}{2} \rceil}^{k} \binom{k}{l} \cdot (1-n(1-c)^2)^{k-l} \cdot (n(1-c)^2)^l \\ &= \mathbb{P}\left[\mathrm{Bin}(k, n(1-c)^2) \geq \frac{k}{2}\right],\end{aligned}$$

where the second inequality uses that the majority string can only be wrong if in at least half of the runs where we observed $j_{n+1} = 1$ there was some error in the remaining string.

Next we use Hoeffding's inequality and obtain, using our assumption $n(1-c)^2 < \frac{1}{2}$, that

$$\begin{aligned}&\mathbb{P}\left[\mathrm{Bin}(k, n(1-c)^2) \geq \frac{k}{2}\right] \\ &= \mathbb{P}\left[\mathrm{Bin}(k, n(1-c)^2) - kn(1-c)^2 \geq \frac{k}{2} - kn(1-c)^2\right] \\ &\leq \exp\left(-k\frac{(1-2n(1-c)^2)^2}{2}\right).\end{aligned}$$

We now set this last expression $\leq \frac{\delta}{2}$ for $\delta \in (0, 1)$ and rearrange the inequality to

$$k \geq \frac{2}{(1-2n(1-c)^2)^2} \ln\left(\frac{2}{\delta}\right). \tag{5.3}$$

Combining Eqs. (5.3) and (5.2) we now require

$$\exp\left(-\frac{2\left(\frac{m}{2} - \frac{2}{(1-2n(1-c)^2)^2}\ln\left(\frac{2}{\delta}\right)\right)^2}{m}\right) \overset{!}{\leq} \frac{\delta}{2}.$$

Rearranging this inequality gives

$$m^2 - 2m\left(\left(\frac{1-2n(1-c)^2}{2}\right)^{-2} - 1\right)\ln\left(\frac{2}{\delta}\right) + \left(\frac{1-2n(1-c)^2}{2}\right)^{-4}\ln^2\left(\frac{2}{\delta}\right) \geq 0.$$

By finding the zeros of this quadratic function, we get to the sufficient sample size

$$m \geq \left(\left(\frac{1-2n(1-c)^2}{2}\right)^{-2} - 1\right)\ln\left(\frac{2}{\delta}\right) + \sqrt{\left(\left(\left(\frac{1-2n(1-c)^2}{2}\right)^{-2} - 1\right)\ln\left(\frac{2}{\delta}\right)\right)^2 - \left(\frac{1-2n(1-c)^2}{2}\right)^{-4}\ln^2\left(\frac{2}{\delta}\right)}.$$

This is in particular guaranteed if

$$m \geq \frac{4}{(1-2n(1-c)^2)^2}\ln\left(\frac{2}{\delta}\right).$$

Note that this lower bound in particular implies $m \geq 2k$, as required earlier in the proof. This proves the claim of the theorem thanks to the union bound. □

Morally speaking, Theorem 4 shows that for product distributions which are close enough to the uniform distribution the sample complexity upper bound is the same as for the unbiased case. We conjecture that there is an explicit noise threshold above which this sample complexity cannot be reached (see the discussion in Sect. 6), but have not yet succeeded in identifying such a critical value.

In this section, we have discussed the case of quantum training data that perfectly represents the target function in a superposition state. Similar results can be proved in the case of noisy quantum training data. As the reasoning is analogous to the one presented here, the details are deferred to "Appendix A."

6 Sample complexity lower bounds

After proving upper bounds on the number of required quantum examples by exhibiting explicit learning procedures in the previous section, we now study the converse

question of sample complexity lower bounds. We will prove both classical and quantum sample complexity lower bounds and then relate them to the above results. Our proof strategy follows a state-discrimination-based strategy from [3].

6.1 Classical sample complexity lower bounds

We first prove a sample complexity lower bound for the classical version of our learning problem that upon comparison with our obtained quantum sample complexity upper bounds shows the advantage of quantum examples over classical training data in this setting. Neither the result nor the proof strategy are new, but we include them for completeness.

Theorem 5 *Let $a \in \{0,1\}^n$, $\mu \in (-1,1)^n$ s.t. μ is c-bounded for some $c \in (0,1]$. Let $\mathcal{A}$ be a classical learning algorithm and let $m \in \mathbb{N}$ be such that upon input of m examples of the form $(x_i, f^{(a)}(x_i))$, with x_i drawn i.i.d. according to D_μ, with probability $\geq 1-\delta$ w.r.t. the choice of training data, $\mathcal{A}$ outputs the string a. Then $m \geq \Omega(n)$.*

Proof Let A be a random variable uniformly distributed on $\{0,1\}^n$. (A describes the underlying string from the initial perspective of the learner.) Let $B = (B_1, \ldots, B_m)$ be a random variable describing the training data corresponding to the underlying string. Our proof will have three main steps: First, we prove a lower bound on $I(A:B)$ from the learning requirement. Second, we observe that $I(A:B) \leq m \cdot I(A:B_1)$. And third, we prove an upper bound on $I(A:B_1)$. Then combining the three steps will lead to a lower bound on m.

We start with the mutual information lower bound. Let $h(B) \in \{0,1\}^n$ denote the random variable describing the output hypothesis of the algorithm $\mathcal{A}$ upon input of training data B. Let $Z = \mathbb{1}_{\{h(B)=A\}}$. By the learning requirement we have $\mathbb{P}[Z=1] \geq 1-\delta$ and thus $H(Z) \leq H(\delta)$. Therefore we obtain

$$\begin{aligned}
I(A:B) &= H(A) - H(A|B) \\
&\geq H(A) - H(A|B,Z) - H(Z) \\
&= H(A) - \mathbb{P}[Z=1]H(A|B,Z=1) - \mathbb{P}[Z=0]H(A|B,Z=0) - H(Z) \\
&\geq n - \mathbb{P}[Z=1]\cdot 0 - \delta n - H(\delta) \\
&= (1-\delta)n - H(\delta) \\
&= \Omega(n).
\end{aligned}$$

We now show that from m examples we can gather at most m times as much information as from a single example. Here we directly cite from [3]. Namely,

$$\begin{aligned}
I(A:B) = H(B) - H(B|A) &= H(B) - \sum_{i=1}^{m} H(B_i|A) \\
&\leq \sum_{i=1}^{m} H(B_i) - H(B_i|A) = \sum_{i=1}^{m} I(A:B_i) = m \cdot I(A:B_1).
\end{aligned}$$

Here, the second step uses independence of the B_i conditioned on A, the third step uses subadditivity of the Shannon entropy, and the final step uses that the distributions of (A, B_i) are the same for all $1 \leq i \leq m$.

We come to the upper bound on the mutual information. Write $B_1 = (X, L)$ for $X \in \{-1, 1\}^n$ and $L \in \{0, 1\}$, i.e., with probability $D_\mu(x)$ we have $(X, L) = (x, f^{(a)}(x))$. Note that $I(A : X) = 0$ because X and A are independent random variables. Also, $I(A : L|X = 1 \ldots 1) = 0$ because $f^{(a)}(1 \ldots 1) = 0 \; \forall a \in \{0, 1\}^n$, and for $x \in \{-1, 1\}^n \setminus \{1 \ldots 1\}$

$$\begin{aligned} I(A : L|X = x) &= I(A_{\{i|X_i=-1\}} : L|X = x) \\ &= H(A_{\{i|X_i=-1\}}|X = x) - H(A_{\{i|X_i=-1\}}|L, X = x) \\ &= |\{i|x_i = -1\}| - (|\{i|x_i = -1\}| - 1) \\ &= 1. \end{aligned}$$

Here, the first step is due to the fact that $f^{(a)}(x)$ does not depend on the entries a_j with $x_j = 1$, the third step follows because $A_{\{i|x_i=-1\}}$ is uniformly distributed on a set of size $2^{|\{i|x_i=-1\}|}$ and $f^{(a)}$ assigns the labels 0 and 1 to half of the elements of that set, respectively.
This now implies

$$\begin{aligned} I(A : B_1) &= I(A : X) + I(A : L|X) \\ &= 0 + \sum_{x \in \{-1,1\}^n} D_\mu(x) I(A : L|X = x) \\ &= 1. \end{aligned}$$

Here, the first step is due to the chain rule for mutual information and the last step simply uses the fact that D_μ defines a probability distribution.
Now we combine our upper and lower bounds on the mutual information and obtain

$$m \geq (1 - \delta)n - H(\delta) = \Omega(n),$$

as claimed. □

Remark 2 The result of Theorem 5 is intuitively clear: In order to identify the underlying string the learning algorithm has to learn n bits of information. However, a condition of the form $f^{(a)}(x) = l$ for $x \in \{0, 1\}^n, l \in \{0, 1\}$, takes away at most one degree of freedom from the initial space $\{0, 1\}^n$ for a and thus from such an equality the algorithm can extract at most 1 bit of information. So at least n examples will be required. This observation is thus neither new nor surprising. But we want to emphasize that this analysis works independently of the product structure of the underlying distribution D_μ.

If we compare the classical lower bound from Theorem 5 with our quantum upper bounds from Theorems 3 and 4 , we conclude that quantum examples allow us to strictly outperform the best possible classical algorithm w.r.t. the number of required examples.

6.2 Quantum sample complexity lower bounds

We can use a similar argument to prove quantum sample complexity lower bounds. Note that steps 1 and 2 carry over with (almost) no changes. Only the analysis of step 3 changes significantly. Even though this proof strategy is possible, as in [3] it can be improved upon by an argument based on state discrimination. We will thus follow this same approach.

An n-independent quantum sample complexity lower bound is given in the following

Lemma 5 *Let* $|\psi_a\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, f^{(a)}(x)\rangle$, $a \in \{0,1\}^n$, $\mu \in (-1,1)^n$ *s.t.* D_μ *is* c*-bounded for some* $c \in (0,1]$. *Let* $\mathcal{A}$ *be a quantum learning algorithm and let* $m \in \mathbb{N}$ *be such that upon input of* m *copies of* $|\psi_a\rangle$, *with probability* $\geq 1-\delta$, $\mathcal{A}$ *outputs the string* a. *Then* $m \geq \Omega(\frac{1}{c}\ln(\frac{1}{\delta}))$.

Remark 3 Note that any quantum sample complexity lower bound will also lower bound the classical sample complexity. Hence, Lemma 2 also holds in the scenario of the previous subsection, which is why we did not discuss the δ-dependence there.

Proof Let $a, b \in \{0,1\}^n$ s.t. there is exactly one $1 \leq i \leq n$ s.t. $a_i \neq b_i$. As $\mathcal{A}$ is able to distinguish the quantum states $|\psi_a\rangle^{\otimes m}$ and $|\psi_b\rangle^{\otimes m}$ with success probability $\geq 1-\delta$, we have $|\langle\psi_a|\psi_b\rangle^m| \leq 2\sqrt{\delta(1-\delta)}$ (see subsection 3.2). We compute

$$\begin{aligned}\langle\psi_a|\psi_b\rangle &= \sum_{x,y\in\{-1,1\}^n} \sqrt{D_\mu(x)D_\mu(y)}\langle x, f^{(a)}(x)|y, f^{(b)}(y)\rangle \\ &= \sum_{x\in\{-1,1\}^n} D_\mu(x)\delta_{f^{(a)}(x),f^{(b)}(x)}.\end{aligned}$$

By our assumption on a and b, $\delta_{f^{(a)}(x),f^{(b)}(x)} \geq \delta_{x_i,1}$. Therefore

$$\langle\psi_a|\psi_b\rangle \geq \mathbb{P}_{D_\mu}[x_i = 1] = \frac{1+\mu_i}{2}.$$

We now combine this with our upper bound and rearrange to obtain

$$\begin{aligned}m &\geq \left(\ln\left(\frac{1+\mu_i}{2}\right)\right)^{-1}\left(\ln(2) + \frac{1}{2}\ln(\delta(1-\delta))\right) \\ &\geq \Omega\left(\frac{1}{\mu_i - 1}\ln(\delta)\right) \\ &\geq \Omega\left(\frac{1}{c}\ln\left(\frac{1}{\delta}\right)\right),\end{aligned}$$

where we used the elementary inequality $\frac{1}{x-1} - \left(\ln\left(\frac{1+x}{2}\right)\right)^{-1} \geq 0$ for $x \in [0,1)$ combined with $\ln(\delta) \leq 0$. □

We will compare this lower bound with our upper bound(s) from Sect. 5 later on. Now we turn to the n-dependent part of the sample complexity lower bound.

Theorem 6 *Let* $|\psi_a\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, f^{(a)}(x)\rangle$, $a \in \{0,1\}^n$, *and* $\mu \in (-1,1)$ *be such that* $\mu_i = \mu \geq 1 - \frac{1}{\ln(n)}$ *for all* $1 \leq i \leq n$. *Let* $\mathcal{A}$ *be a quantum learning algorithm and let* $m \in \mathbb{N}$ *be such that upon input of* m *copies* $|\psi_a\rangle$, *with probability* $\geq 1-\delta$, $\mathcal{A}$ *outputs the string* a, *for* $0 < \delta \leq \frac{1}{3}$. *Then* $m \geq \Omega(\ln(n))$.

Before going into the detailed proof, we give an overview over its underlying idea. The learning assumption implies that $\mathcal{A}$ is able to identify a state from the ensemble $\mathcal{E} = \left\{\left(\frac{1}{2^n}, |\psi_a\rangle^{\otimes m}\right)\right\}_{a\in\{0,1\}^n}$ with success probability $\geq 1-\delta$. Thus we will obtain a lower bound on m by proving an upper bound on the optimal success probability for this state identification task.

Recall that by Theorem 1, the optimal success probability can be upper bounded by the square root of the PGM success probability. Moreover, by Lemma 3, the latter can be computed via the Gram matrix of the ensemble. Thus, we now first study the Gram matrix and its square root and then use these results to bound the optimal success probability.

We first recall a well-known result on the diagonalization of matrices with a specific structure, namely matrices whose entries can be written as Boolean function of the sum of the indices.

Lemma 6 *Let* $G \in \mathbb{R}^{2^n\times 2^n}$ *be a matrix with entries given by* $G(a,b) = g(a+b)$ *for* $a, b \in \{0,1\}^n$ *and a function* $g : \{0,1\}^n \to \mathbb{R}$. *Then*

$$(HGH^{-1})(a,b) = 2^n \hat{g}(a)\delta_{a,b},$$

with $H \in \mathbb{R}^{2^n\times 2^n}$ *given by* $H(a,b) = \frac{(-1)^{a\cdot b}}{\sqrt{2^n}}$. *In other words, the set of eigenvalues of* G *is given by* $\{2^n\hat{g}(a) \mid a \in \{0,1\}^n\}$ *and* G *is unitarily diagonalized by* H.

Proof The proof can be found in [3], we reproduce it in "Appendix B" □

We will later apply this result for G being the Gram matrix corresponding to the ensemble in our state identification task. Motivated by Lemma 3, we first use the diagonalization of such a matrix to explicitly compute the diagonal entries of the matrix square root.

Corollary 1 *Let* $G \in \mathbb{R}^{2^n\times 2^n}$ *be a matrix with entries given by* $G(a,b) = g(a+b)$ *for* $a, b \in \{0,1\}^n$ *and a function* $g : \{0,1\}^n \to \mathbb{R}$. *Then, for every* $a \in \{0,1\}^n$

$$\sqrt{G}(a,a) = \frac{1}{\sqrt{2^n}} \sum_{j\in\{0,1\}^n} \sqrt{\hat{g}(j)}.$$

Proof The proof can be found in [3], we reproduce it in "Appendix B." □

With this, we can now prove Theorem 6:

Proof of Theorem 6 As discussed above, we consider the problem of state identification with the ensemble $\mathcal{E} = \left\{\left(\frac{1}{2^n}, |\psi_a\rangle^{\otimes m}\right)\right\}_{a\in\{0,1\}^n}$. By Lemma 3, with the Gram matrix $G_m(a,b) := \frac{1}{2^n}\langle\psi_a|\psi_b\rangle^m$ we can write the success probability as

$$P^{PGM}(\mathcal{E}) = \sum_{a\in\{0,1\}^n} \sqrt{G_m}(a,a)^2.$$

In our scenario, the Gram matrix has entries

$$\begin{aligned} G_m(a,b) &= \frac{1}{2^n}\langle\psi_a|\psi_b\rangle^m \\ &= \frac{1}{2^{n+m}}\left(1+\mu^{\mathrm{d}_H(a,b)}\right)^m = \frac{1}{2^{n+m}}\left(1+\mu^{\mathrm{d}_H(a+b,0)}\right)^m. \end{aligned}$$

This can, e.g., be shown by induction on n when observing that

$$\begin{aligned} &\mathbb{P}_{D_\mu}[f^{(a)}(x) = f^{(b)}(x)] \\ &= \mathbb{P}_{D_\mu}\left[f^{(a_{1:n-1})}(x_{1:n-1}) = f^{(b_{1:n-1})}(x_{1:n-1}) \ \wedge\ a_n\frac{1-x_n}{2} = b_n\frac{1-x_n}{2}\right] \\ &\quad + \mathbb{P}_{D_\mu}\left[f^{(a_{1:n-1})}(x_{1:n-1}) \neq f^{(b_{1:n-1})}(x_{1:n-1}) \ \wedge\ a_n\frac{1-x_n}{2} \neq b_n\frac{1-x_n}{2}\right]. \end{aligned}$$

In particular, we can write $G_m(a,b) = f_m(a+b)$ for the function $f_m(x) = \frac{1}{2^{n+m}}\left(1+\mu^{\mathrm{d}_H(x,0)}\right)^m$. From now on, we will write $|x| := \mathrm{d}_H(x,0)$. By Corollary 1, we can upper bound the diagonal entries of $\sqrt{G_m}$ (and thus the PGM and the optimal success probability) by upper bounding the (unbiased) Fourier coefficients of f_m. To this end, consider for $j \in \{0,1\}^n$

$$\begin{aligned} 0 \le \hat{f}_m(j) &= \mathbb{E}_{z\sim U(\{0,1\}^n)}\left[\frac{1}{2^{n+m}}\left(1+\mu^{|z|}\right)^m(-1)^{j\cdot z}\right] \\ &= \frac{1}{2^{n+m}}\sum_{L=0}^{m}\binom{m}{L}\mathbb{E}_{z\sim U(\{0,1\}^n)}\left[\mu^{L|z|}(-1)^{j\cdot z}\right]. \end{aligned}$$

We now rewrite the expectations on the right-hand side

$$\begin{aligned} &\mathbb{E}_{z\sim U(\{0,1\}^n)}\left[\mu^{L|z|}(-1)^{j\cdot z}\right] \\ &= \frac{1}{2^n}\sum_{\ell=0}^{n}\sum_{k=\max\{0,\ell-(n-|j|)\}}^{\min\{\ell,|j|\}}\binom{|j|}{k}\binom{n-|j|}{\ell-k}(-1)^k\mu^{L\cdot\ell} \\ &= \frac{1}{2^n}\sum_{k=0}^{|j|}\binom{|j|}{k}(-1)^k\sum_{\ell=k}^{k+n-|j|}\binom{n-|j|}{\ell-k}\mu^{L\cdot\ell} \end{aligned}$$

$$= \frac{1}{2^n} \sum_{k=0}^{|j|} \binom{|j|}{k} (-1)^k \mu^{L\cdot k} \underbrace{\sum_{\ell=0}^{n-|j|} \binom{n-|j|}{\ell} \mu^{L\cdot \ell}}_{=(1+\mu^L)^{n-|j|}}$$

$$= \frac{\left(1+\mu^L\right)^{n-|j|}}{2^n} \underbrace{\sum_{k=0}^{|j|} \binom{|j|}{k} (-1)^k \mu^{L\cdot k}}_{=(1-\mu^L)^{|j|}}$$

$$= \frac{\left(1+\mu^L\right)^{n-|j|} \left(1-\mu^L\right)^{|j|}}{2^n}.$$

This allows us to upper bound the Fourier coefficients of f as follows:

$$\begin{aligned}
\hat{f}_m(j) &= \frac{1}{2^{n+m}} \sum_{L=0}^{m} \binom{m}{L} \left(\frac{1+\mu^L}{2}\right)^{n-|j|} \left(\frac{1-\mu^L}{2}\right)^{|j|} \\
&\leq \frac{1}{2^{n+m}} \sum_{L=0}^{m} \binom{m}{L} \left(\frac{1+\mu}{2}\right)^{n-|j|} \left(\frac{1-\mu^m}{2}\right)^{|j|} \\
&= \frac{1}{2^n} \left(\frac{1+\mu}{2}\right)^{n-|j|} \left(\frac{1-\mu^m}{2}\right)^{|j|}.
\end{aligned}$$

According to Lemma 6, this now gives us the following upper bound on the diagonal entries of the root of the Gram matrix

$$\begin{aligned}
\sqrt{G_m}(a,a) &\leq \frac{1}{2^n} \sum_{j\in\{0,1\}^n} \sqrt{\left(\frac{1+\mu}{2}\right)^{n-|j|} \left(\frac{1-\mu^m}{2}\right)^{|j|}} \\
&= \frac{1}{2^n} \sum_{k=0}^{n} \binom{n}{k} \sqrt{\left(\frac{1+\mu}{2}\right)^{n-k} \left(\frac{1-\mu^m}{2}\right)^{k}} \\
&= \frac{1}{2^n} \left(\sqrt{\frac{1+\mu}{2}} + \sqrt{\frac{1-\mu^m}{2}}\right)^n,
\end{aligned}$$

and this in turn allows us to bound the PGM success probability as

$$\begin{aligned}
P^{PGM}(\mathcal{E}) &= \sum_{a\in\{0,1\}^n} \sqrt{G_m}(a,a)^2 \\
&\leq \frac{1}{2^n} \left(\sqrt{\frac{1+\mu}{2}} + \sqrt{\frac{1-\mu^m}{2}}\right)^{2n} \\
&= \left(\frac{1}{2}\left(\sqrt{1+\mu} + \sqrt{1-\mu^m}\right)\right)^{2n}.
\end{aligned}$$

We combine this with our learning requirement and Theorem 1 to obtain

$$1-\delta \leq P^{opt}(\mathcal{E}) \leq \sqrt{P^{PGM}(\mathcal{E})} \leq \left(\frac{1}{2}\left(\sqrt{1+\mu}+\sqrt{1-\mu^m}\right)\right)^n.$$

This can be rearranged (using $\delta < \frac{1}{3}$) to

$$m = \frac{-\log\left(1-\left(2\cdot\sqrt[n]{1-\delta}-\sqrt{1+\mu}\right)^2\right)}{\log\frac{1}{\mu}}.$$

With $\log(1+x) \leq x$ we obtain $\frac{1}{\log\frac{1}{\mu}} \geq \frac{1}{\frac{1}{\mu}-1} = \frac{\mu}{1-\mu}$ and

$$-\log\left(1-\left(2\cdot\sqrt[n]{1-\delta}-\sqrt{1+\mu}\right)^2\right) \geq \left(2\cdot\sqrt[n]{1-\delta}-\sqrt{1+\mu}\right)^2.$$

For $\mu \geq 1-\frac{1}{\ln(n)}$ we now obtain (for n large enough)

$$m \geq (\ln(n)-1)\cdot\left(2\sqrt{\frac{2}{3}}-\sqrt{2}\right) = \Omega\left(\ln(n)\right),$$

and this finishes the proof. □

Note that this proof strategy also yields for a strictly increasing function $g : \mathbb{N} \to \mathbb{R}_{>0}$ with $\lim_{n\to\infty} g(n) = \infty$ and for a distribution D_μ with $\mu_i \geq 1-\frac{1}{g(n)}$ for all $1 \leq i \leq n$ the sample complexity lower bound $\Omega(g(n))$ (for n large enough). This is consistent with the intuition that solving the learner problem becomes harder when the distribution is more strongly biased towards the uninformative instance with all entries equal to 1.

We now compare this lower bound to our previously obtained upper bounds. First, we consider the n-independent part of the bounds. When comparing Theorem 3 with Lemma 5, we obtain

$$\Omega\left(\frac{1}{c}\ln\left(\frac{1}{\delta}\right)\right) \leq m \leq \mathcal{O}\left(\left(\ln\left(\frac{1}{1-c+\frac{c^2}{2}}\right)\right)^{-1}\ln\left(\frac{1}{\delta}\right)\right).$$

We study this for $\delta \ll 1$ (high confidence) and $c \ll 1$ (high bias). Then Taylor expansion shows

$$\left(\ln\left(\frac{1}{1-c+\frac{c^2}{2}}\right)\right)^{-1} = \frac{1}{c}+\frac{c}{6}+\mathcal{O}(c^2) \quad \text{for } c \ll 1.$$

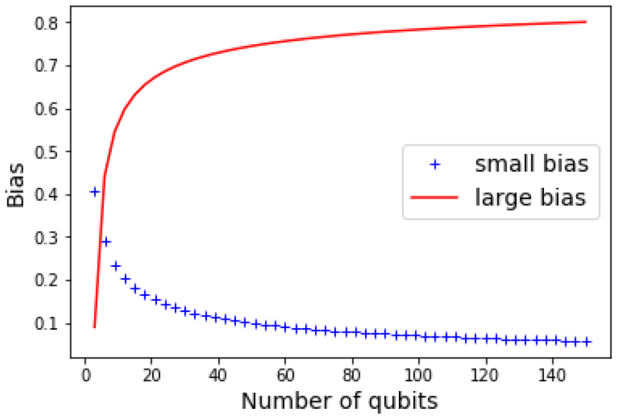

Fig. 1 A plot comparing the maximal bias allowed in Theorem 4 (depicted by the blue crosses) with the minimal bias required in Theorem 6 (depicted by the red line) (Color figure online)

Hence, lower and upper bounds coincide in the relevant region for δ and c, so the n-independent part of the sample complexity upper bound provided by Algorithm 3 is optimal.

However, in comparing Theorem 4 with Lemma 5 we see a discrepancy between lower and upper bound for the relevant region $\delta \ll 1$ and $c-(1-\frac{1}{\sqrt{2n}}) \ll 1$. Therefore we conjecture that the c-dependence of the upper bound arising from Theorem 4 is not optimal.

Now we compare the bounds w.r.t. the n-dependence, i.e., we compare Theorem 3 with Theorem 6, and obtain

$$\Omega\left(\ln(n)\right) \leq m \leq \mathcal{O}\left(\frac{1}{c}\ln(n)\right).$$

But in Theorem 6, we assumed that $\mu_i \geq 1-\frac{1}{\ln(n)}$ for all $1 \leq i \leq n$. When considering values for μ lying on this threshold, we can rephrase this as condition on the (then n-dependent) c-boundedness parameter, namely $c \leq \frac{1}{\ln(n)}$. So when honestly including the n-dependence of c, our comparison becomes

$$\Omega\left(\ln(n)\right) \leq m \leq \mathcal{O}\left(\ln^2(n)\right)$$

and is thus not tight.

Finally, we want to point towards a second unsatisfactory aspect of our results. We provide an n-dependent quantum sample complexity lower bound for "large" noise and an n-independent quantum sample complexity upper bound for "small" noise. However, there is a large discrepancy between the obtained characterizations of "small" and "large" noise. That this already becomes relevant for moderate n can be seen in Fig. 1.

Hence, we did not succeed in identifying a bias threshold beyond which the sample complexity qualitatively differs from the unbiased case, but merely provided a region in

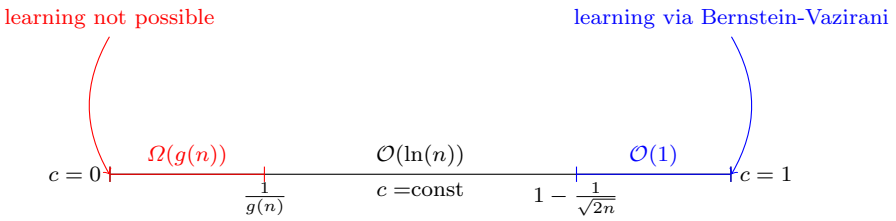

Fig. 2 Overview of the quantum sample complexity upper and lower bounds from Theorems 3, 4 and 6 depending on the c-boundedness parameter (without noise in the training data). Here, $g : \mathbb{N} \to \mathbb{R}_{>0}$ is a strictly increasing function with $\lim_{n\to\infty} g(n) = \infty$ (Color figure online)

which such a threshold would lie. To improve upon our results, it would be necessary to modify either the proof of Theorem 4 to allow for stronger bias or the proof of Theorem 6 to allow for weaker bias. In particular, it would be interesting to obtain a non-trivial quantum sample complexity lower bound for constant bias, i.e., without introducing n-dependence into the c-boundedness parameter. However, we currently do not see whether our proof strategies admit such an improvement.

7 Conclusion and outlook

In this paper, we extended a well-known quantum learning strategy for linear functions from the uniform distribution to biased product distributions. This approach naturally led to a distinction between a procedure for arbitrary (not full) bias and a procedure for small bias, the latter with a significantly better performance. Moreover, we showed that the second procedure is (to a certain degree) stable w.r.t. noise in the training data and in the performed quantum gates. Finally, we also provided lower bounds on the size of the training data required for the learning problem, both in the classical and in the quantum setting. The sample complexity upper and lower bounds in the case of no noise are summarized in Fig. 2.

We want to conclude by outlining some open questions for future work:

- Can we identify a bias threshold s.t. the optimal sample complexity below the threshold differs qualitatively from the one above it?
- Is our learning procedure for small bias also stable w.r.t. different types of noise in the training data, e.g., malicious noise?
- Our explicit learning algorithms also give upper bounds on the computational complexity of our learning problem. Can we find corresponding lower bounds to facilitate a discussion of optimality w.r.t. runtime?
- Can we find more examples of learning tasks (i.e., function classes) where quantum training data yields an advantage w.r.t. sample and/or time complexity?

Acknowledgements Open Access funding provided by Projekt DEAL. First, I want to thank my supervisor Michael Wolf for several stimulating discussions concerning questions of quantum learning. Also, I want to thank Benedikt Graswald for proofreading a first draft of this paper and for his constructive comments. Finally, I am grateful to Andrea Rocchetto for useful comments to improve the result of "Appendix A.3" and for suggesting further references. Also, I thank the reviewers for their constructive criticism. Support

from the TopMath Graduate Center of the TUM Graduate School at the Technische Universität München, Germany, and from the TopMath Program at the Elite Network of Bavaria is gratefully acknowledged.

Compliance with ethical standards

Conflict of interest The author declares that he has no conflict of interest.

Appendix

A Stability w.r.t. noise

Both algorithms presented in Sect. 5 implicitly assume that the quantum example state perfectly represents the underlying function and that all quantum gates performed during the computation are perfectly accurate. In this section, we relax these assumptions. We will do so separately, but our analysis shows that moderate noise in the training data and moderately faulty quantum gates can be tolerated at the same time.

A.1 Noisy training data

One of the most well-studied noise models in classical learning theory is that of random classification noise. Here, the training data are assumed to be s.t. with probability $1-\eta$, the learning algorithm obtains a correct example, and with probability η, the examples label is flipped. In [4], this is translated to a quantum example state which in our notation has the form

$$|\varphi_a^{\text{noisy}}\rangle = \sqrt{1-\eta}\left(\sum_{x\in\{-1,1\}}\sqrt{D_\mu(x)}|x,\, f^{(a)}(x)\rangle\right) + \sqrt{\eta}\left(\sum_{x\in\{-1,1\}}\sqrt{D_\mu(x)}|x,\, f^{(a)}(x)\oplus 1\rangle\right).$$

We will only shortly comment on how to battle this type of noise with our learning strategy at the end of this subsection. Instead, our focus will be on a performance analysis of our algorithm in the case of noisy training data similar to [13]. This means that we now assume our quantum example state to be of the form

$$|\psi_a^{\text{noisy}}\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, \sum_{i=1}^n a_i \frac{1-x_i}{2} + \xi_{x_i}^i\rangle,$$

where the $\xi_{x_i}^i$, for $1 \leq i \leq n$ and $x_i \in \{-1, 1\}$, are independent random variables distributed according to Bernoulli distributions with parameters η^i (i.e., $\mathbb{P}[\xi_{x_i}^i = 1] = \eta^i = 1 - \mathbb{P}[\xi_{x_i}^i = 0]$ for all $1 \leq i \leq n$) and addition is understood modulo 2.

Here, we choose a noise model that is rather general but we make an important restriction. Namely, we do not allow a noise ξ_x that depends in an arbitrary way on x but rather we require the noise to have a specific sum structure $\xi_x = \sum_{i=1}^n \xi_{x_i}^i$. This requirement will later imply that also the noisy Fourier coefficients factorize. As this factorization is crucial for our analysis, with our strategy we cannot generalize the results of [13] on that more general noise model.

We first examine the result of applying the same procedure as in Algorithm 2 to a copy of a noisy quantum example state $|\psi_a^{\text{noisy}}\rangle$. To simplify referencing, we write this down one more time as Algorithm 5 even though the procedure is exactly the same, only the form of the input changes.

Algorithm 5 Generalized Bernstein–Vazirani algorithm with noisy training data

Input: $|\psi_a^{\text{noisy}}\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, \sum_{i=1}^n a_i \frac{1-x_i}{2} + \xi_{x_i}^i\rangle$, as well as $\mu \in [-1, 1]$
Output: See Theorem 7
Success Probability: $\frac{1}{2}$.

1: Perform the μ-biased QFT H_μ on the first n qubits, obtain the state $(H_\mu \otimes \mathbb{1})|\psi_a^{\text{noisy}}\rangle$.
2: Perform a Hadamard gate on the last qubit, obtain the state $(H_\mu \otimes H)|\psi_a^{\text{noisy}}\rangle$.
3: Measure each qubit in the computational basis and observe outcome $j = j_1 \dots j_{n+1}$.
4: **if** $j_{n+1} = 0$ **then** ▷ This corresponds to a failure of the algorithm.
5: Output $o = \perp$.
6: **else if** $j_{n+1} = 1$ **then** ▷ This corresponds to a success of the algorithm.
7: Output $o = j_1 \dots j_n$.
8: **end if**

Similarly to our previous analysis, we will first study the Fourier coefficients that are relevant for the sampling process in Algorithm 5.

Lemma 7 *Let $a \in \{0, 1\}^n$. Let $\xi_{x_i}^i$, for $1 \leq i \leq n$ and $x_i \in \{-1, 1\}$, be independent Bernoulli distributions, let $g^{(a)}(x) := (-1)^{\sum_{i=1}^n a_i \frac{1-x_i}{2} + \xi_{x_i}^i}$ and let $\mu \in (-1, 1)$. Then the μ-biased Fourier coefficients of $g^{(a)}$ satisfy: For $y \in \{0, 1\}^n$, with probability*

$$\prod_{l=1}^n \left(y_l \cdot 2\eta^l(1-\eta^l) + (1-y_l)\cdot(1-2\eta^l(1-\eta^l))\right),$$

it holds that

$$\hat{g}_{\mu}^{(a)}(j) = \prod_{l:a_l=0} \left(y_l \cdot (-1)^{b_l} \left((1-j_l)\mu_l + j_l\sqrt{1-\mu_l^2} \right) + (1-y_l) \cdot (-1)^{b_l}(1-j_l) \right) \cdot \prod_{l:a_l=1} \left(y_l \cdot (-1)^{b_l}(1-j_l) + (1-y_l) \cdot (-1)^{b_l} \left((1-j_l)\mu_l + j_l\sqrt{1-\mu_l^2} \right) \right).$$

Proof The proof is analogous to the one of Lemma 4, see "Appendix B." □

We now make a step analogous to the one from Lemma 4 to Theorem 2 in order to understand the output of Algorithm 5.

Theorem 7 *Let* $|\psi_a^{noisy}\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, \sum_{i=1}^n a_i \frac{1-x_i}{2} + \xi_{x_i}^i\rangle$ *be a noisy quantum example state,* $a \in \{0,1\}^n$, $\mu \in (-1,1)^n$. *Then Algorithm 5 provides an outcome* $|j_1 \dots j_{n+1}\rangle$ *with the following properties:*

(i) $\mathbb{P}[j_{n+1} = 0] = \frac{1}{2} = \mathbb{P}[j_{n+1} = 1]$.
(ii) *For any* $1 \le i \le n$, *with probability* $1 - 2\eta^i(1-\eta^i)$ *it holds that*

$$\mathbb{P}[a_i = 0 \neq j_i \,|\, j_{n+1} = 1] = 0, \quad \mathbb{P}[a_i = 1 \neq j_i \,|\, j_{n+1} = 1] = \mu^2.$$

(iii) *For any* $1 \le i \le n$, *with probability* $2\eta^i(1-\eta^i)$ *it holds that*

$$\mathbb{P}[a_i = 0 \neq j_i \,|\, j_{n+1} = 1] = 1 - \mu^2, \quad \mathbb{P}[a_i = 1 \neq j_i \,|\, j_{n+1} = 1] = 1.$$

Note that in the scenario of Theorem 7 the underlying distribution D_μ is known to the algorithm as μ is provided as part of the input (see Algorithm 5). Building on this subroutine, we will now describe an amplified procedure for moderate noise (which is made precise in Theorem 8) in Algorithm 6 analogous to the one described in Sect. 5.2. Again, only the input changes, but we write the procedure down explicitly to simplify referencing.

Theorem 8 *Let* $|\psi_a^{noisy}\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, \sum_{i=1}^n a_i \frac{1-x_i}{2} + \xi_{x_i}^i\rangle$, *with* $a \in \{0,1\}^n$, $\mu \in (-1,1)^n$ *s.t.* D_μ *is* c*-bounded for some* $c \in (0,1]$ *satisfying* $c > 1 - \frac{1}{2\sqrt{n}}$. *Further assume that* $2\eta^i(1-\eta^i) < \frac{1}{5n}$ *for all* $1 \le i \le n$, *write* $\rho := \max_{1\le i\le n} 2\eta^i(1-\eta^i)$. *Then* $\mathcal{O}\left(\max\left\{\frac{1}{(1-5n\rho)^2}, \frac{1}{(1-4n(1-c)^2)^2}\right\} \ln\left(\frac{1}{\delta}\right)\right)$ *copies of the quantum example state* $|\psi_a\rangle$ *suffice to guarantee that with probability* $\ge 1-\delta$ *Algorithm 6 outputs the string* a.

As in Theorem 4, our restrictions on both the c-boundedness parameter and the noise strength lead to a basically n-independent sample complexity upper bound.

Proof The proof is analogous to the one of Theorem 4, see "Appendix B." □

Algorithm 6 Amplified Generalized Bernstein–Vazirani algorithm with noisy training data

Input: m copies of $|\psi_a^{\text{noisy}}\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, \sum_{i=1}^n a_i \frac{1-x_i}{2} + \xi^i_{x_i}\rangle$ for $a \in \{0,1\}^n$, where the number of copies is $m \geq C\left(\max\left\{\frac{1}{(1-5n\rho)^2}, \frac{1}{(1-4n(1-c)^2)^2}\right\}\ln\left(\frac{1}{\delta}\right)\right)$, as well as $\mu \in [-1,1]^n$ and $c \in (0,1]$ s.t. D_μ is c-bounded.
Output: $a \in \{0,1\}^n$
Success Probability: $\geq 1-\delta$

1: **for** $1 \leq l \leq m$ **do**
2: Run Algorithm 5 on the l^{th} copy of $|\psi_a^{\text{noisy}}\rangle$, store the output as $o^{(l)}$.
3: **end for**
4: **if** $\exists 1 \leq l \leq m : o^{(l)} \neq \perp$ **then**
5: **for** $1 \leq i \leq n$ **do**
6: Let $o_i = \arg\max_{r\in\{0,1\}}|\{1 \leq l \leq m | o_i^{(l)} = r\}|$.
7: **end for**
8: Output $o = o_1 \ldots o_n$.
9: **else if** $\forall 1 \leq l \leq m : o^{(l)} = \perp$ **then**
10: Output $o = \perp$.
11: **end if**

The previous Theorem shows that if the bias is not too strong and if the noise is not too random (i.e., the probability of adding a random 1 is either very low or very high), then learning is possible with essentially the same sample complexity as in the case without noise (compare Theorem 4).
Note that the proof of Theorem 8 shows that the exact choices of the bounds (in our formulation $c > 1 - \frac{1}{2\sqrt{n}}$ and $2\eta^i(1-\eta^i) < \frac{1}{5n}$) are flexible to some degree with a trade-off. If we have a better bound on c, we can loosen our requirement on the η^i and vice versa.

Also observe that the requirement of "not too random noise" is natural. If $2\eta^i(1-\eta^i) \to \frac{1}{2}$ or, equivalently, $\eta^i \to \frac{1}{2}$, then the label in the noisy quantum example state becomes completely random and thus no information on the string a can be extracted from it. Our bound gives a quantitative version of this intuition.

Nevertheless, the restriction which we put on the noise can be considered quite strong because of its n-dependence. This can, however, be relaxed at the cost of a looser sample complexity upper bound. Namely, similarly to the difference between the proofs of Theorems 3 and 4 , if we, e.g., only assume $2\eta^i(1-\eta^i) < \frac{1}{5}$ for all $1 \leq i \leq n$, we can first for each coordinate separately bound the probability of the noise variables becoming relevant in at least $\frac{k}{5}$ runs using Hoeffding's inequality and then use the union bound. This will yield a quantum sample complexity upper bound with an n-dependent term of the form $\ln(n)$. Hence, if we assume a c-boundedness parameter strongly restricted as in Theorems 4 or 8, but obtain faulty training data states without an n-dependent noise bound as in Theorem 8, then we can still obtain a sample complexity upper bound with the same n-dependence as in Theorem 3.

Finally, as promised at the beginning of this subsection, we shortly describe how to use the ideas presented in this subsection in the case of random classification noise as

in [4]. If the quantum learning algorithm has access to copies of a quantum example state

$$|\varphi_a^{\text{noisy}}\rangle = \sqrt{1-\eta}\left(\sum_{x\in\{-1,1\}}\sqrt{D_\mu(x)}|x,\, f^{(a)}(x)\rangle\right) + \sqrt{\eta}\left(\sum_{x\in\{-1,1\}}\sqrt{D_\mu(x)}|x,\, f^{(a)}(x)\oplus 1\rangle\right),$$

then we observe that applying the μ-biased Fourier transform to SSthe first n qubits and the standard Fourier transform to the last qubit gives

$$(H_\mu^{\otimes n}\otimes H)\left(|\varphi_a^{\text{noisy}}\rangle\right) = \frac{\sqrt{1-\eta}+\sqrt{\eta}}{\sqrt{2}}|0,\ldots,0\rangle + \frac{\sqrt{1-\eta}-\sqrt{\eta}}{\sqrt{2}}\sum_{j\in\{0,1\}}\hat{g}_\mu(j)|j,1\rangle.$$

Hence, compared to the scenario studied in section 5 the probabilities of observing a certain string as measurement outcome are simply scaled by a factor of $(\sqrt{1-\eta}\pm\sqrt{\eta})^2 = 1\pm 2\sqrt{\eta(1-\eta)}$. So our analysis carries over almost directly. We do not give the detailed reasoning here but only mention that incorporating the now rescaled probabilities basically changes the sample complexity upper bounds from the non-noisy case by a factor of $\frac{1}{(\eta-\frac{1}{2})^2}$, which is again in accordance with the intuition that the learning task becomes hard—and eventually impossible—for $\eta\to\frac{1}{2}$.

A.2 Faulty quantum gates

We now turn to the (more realistic) setting where the quantum gates in our computation (i.e., the μ-biased quantum Fourier transforms) are not implemented exactly but only approximately. In this scenario, we obtain

Lemma 8 *Let $|\psi_a\rangle = \sum_{x\in\{-1,1\}^n}\sqrt{D_\mu(x)}|x,\, f^{(a)}(x)\rangle$ be a quantum example state, with $a\in\{0,1\}^n$, $\mu\in(-1,1)^n$. Then a version of Algorithm 2 with H_μ replaced by $H_{\tilde{\mu}}$ for $\left\|H_\mu - H_{\tilde{\mu}}\right\|_2 \le \varepsilon$ provides an outcome $|j_1\ldots j_{n+1}\rangle$ with the following properties:*

(i) $|\mathbb{P}[j_{n+1}=0]-\frac{1}{2}|\le\varepsilon$ *and* $|\mathbb{P}[j_{n+1}=1]-\frac{1}{2}|\le\varepsilon$,
(ii) $|\mathbb{P}[j_1\ldots j_n = a|j_{n+1}=1] - \prod_{l:a_l=1}(1-\mu_l^2)|\le\varepsilon$,
(iii) *for $c\neq a$:*

$$\left|\mathbb{P}[j_1\ldots j_n = c|j_{n+1}=1] - \prod_{l:a_l=0}(1-c_l)\cdot\prod_{l:a_l=1}\left((1-c_l)\mu_l^2 + c_l(1-\mu_l^2)\right)\right|\le\varepsilon,$$

(iv) $\mathbb{P}[\exists 1 \leq i \leq n : a_i = 0 \neq j_i | j_{n+1} = 1] \leq \varepsilon$, *and*

(v) $\mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq j_i | j_{n+1} = 1] \leq \sum_{i=1}^{n} \mu_i^2 + \varepsilon$. *In particular, if* D_μ *is* c*-bounded, then* $\mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq j_i | j_{n+1} = 1] \leq n(1-c)^2 + \varepsilon$.

Proof This follows from Theorem 2 because the outcome probabilities are the squares of the amplitudes, and thus, the difference in outcome probabilities can be bounded by the 2-norm of the difference of the quantum states after applying the biased quantum Fourier transform and its approximate version. □

Now we can proceed analogously to the proof strategy employed in Theorem 8 to derive

Theorem 9 *Let* $|\psi_a\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle$, $a \in \{0,1\}^n$, $\mu \in (-1,1)^n$ *s.t.* D_μ *is* c*-bounded for some* $c \in (0,1]$ *satisfying* $c > 1 - \sqrt{\frac{1-2\varepsilon}{2n}}$. *Then*

$$\mathcal{O}\left(\max\left\{\frac{1}{(1-2\varepsilon)^2}, \frac{1}{1-2(n(1-c)^2+\varepsilon)^2}\right\} \ln\left(\frac{1}{\delta}\right) + \varepsilon\right)$$

copies of the quantum example state $|\psi_a\rangle$ *suffice to guarantee that, with probability* $\geq 1-\delta$, *a version of Algorithm 4 with* H_μ *replaced by* $H_{\tilde{\mu}}$ *for* $\left\|H_\mu - H_{\tilde{\mu}}\right\|_2 \leq \varepsilon \in (0, \frac{1}{2})$ *outputs the string* a.

In particular, the sample complexity upper bound from Theorem 4 remains basically untouched if quantum gates with small error are used.

A.3 The case of unknown underlying distributions

An interesting consequence of the result of the previous subsection is the possibility to drop the assumption of prior knowledge of the underlying product distribution, as was already observed in [17] for a similar scenario. The important observations towards this end are given in this subsection.

Lemma 9 (Lemma 5 in [17])

Let $A = A_n \cdots A_1$ *be a product of unitary operators* A_j. *Assume that for every* A_j *there exists an approximation* $\tilde{A}_j$ *s.t.* $\left\|A_j - \tilde{A}_j\right\| \leq \varepsilon_j$. *Then it holds that*

$$\left\|A_n \cdots A_1 - \tilde{A}_n \cdots \tilde{A}_1\right\| \leq \sum_{j=1}^{n} \varepsilon_j,$$

i.e., the operator $\tilde{A} := \tilde{A}_n \cdots \tilde{A}_1$ *is an* ε*-approximation to* A *w.r.t. the operator norm.*

Proof This can be proven by induction using the triangle inequality and the fact that a unitary operator has operator norm equal to 1. For details, the reader is referred to [17]. □

This can be used to derive (compare again [17])

Corollary 2 *Let $\mu \in (-1,1)^n$ be s.t. the distribution D_μ is c-bounded for $c \in (0,1]$. Let $\tilde{\mu} \in (-1,1)^n$ satisfy $\|\mu - \tilde{\mu}\|_\infty \leq \varepsilon$. Then the corresponding biased quantum Fourier transforms satisfy*

$$\left\| H_\mu - H_{\tilde{\mu}} \right\| \leq 2\sqrt{2} n \gamma \varepsilon,$$

where $\gamma = \frac{1}{c^2}\left((2-c)\frac{3}{2\sqrt{2}c} + 1\right)$.

Proof This proof is given in "Appendix B." □

The next Lemma is on approximating the bias parameter of an unknown product distribution from examples. (Compare the closing remark in Appendix A of [17].)

Lemma 10 *Using $m \leq \mathcal{O}(\frac{8\gamma^2 \cdot n^2}{\varepsilon^2} \ln(\frac{n}{\delta}))$ copies of the quantum example state $|\psi_a\rangle$ (or of $|\psi_a^{noisy}\rangle$) for a product distribution D_μ with bias vector $\mu \in (-1,1)^n$ s.t. D_μ is c-bounded for $c \in (0,1]$ one can, with probability $\geq 1-\delta$, output $\tilde{\mu} \in (-1,1)^n$ s.t. $\left\| H_\mu - H_{\tilde{\mu}} \right\| \leq \varepsilon$.*

Proof Recall that $\mu_i = \mathbb{E}_{D_\mu}[x_i]$. Via a standard application of Hoeffding's inequality we conclude that $\mathcal{O}(\frac{8\gamma^2 \cdot n^2}{\varepsilon^2} \ln(\frac{1}{\delta}))$ examples drawn i.i.d. from D_μ (which can be obtained from copies of the quantum example state by measuring the corresponding subsystem) are sufficient to guarantee that, with probability $\geq 1-\delta$, the empirical estimate $\hat{\mu}_i$ satisfies $|\mu_i - \hat{\mu}_i| \leq \frac{\varepsilon}{2\sqrt{2}\gamma \cdot n}$. As each component of a copy of the quantum example state can be measured separately, we see —using the union bound, that $\mathcal{O}(\frac{8\gamma^2 \cdot n^2}{\varepsilon^2} \ln(\frac{n}{\delta}))$ copies of the (possibly noisy) quantum example state suffice to guarantee that, with probability $\geq 1-\delta$, it holds that $\left\| \mu - \hat{\mu} \right\|_\infty \leq \frac{\varepsilon}{2\sqrt{2}\gamma \cdot n}$. Now we can apply the previous Corollary to finish the proof. □

If we now combine this result with Theorem 9, we obtain a sample complexity upper bound for our learning problem without assuming the underlying distribution to be known in advance.

Corollary 3 *Let $|\psi_a\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)}|x, f^{(a)}(x)\rangle$, $a \in \{0,1\}^n$, $\mu \in (-1,1)^n$ s.t. D_μ is c-bounded for some $c \in (0,1]$ satisfying $c > 1 - \sqrt{\frac{1-2\varepsilon}{2n}}$. Then there exists a quantum algorithm which, given access to*

$$\mathcal{O}\left(\frac{8\gamma^2 \cdot n^2}{\varepsilon^2} \ln\left(\frac{n}{\delta}\right) + \max\left\{\frac{1}{(1-2\varepsilon)^2}, \frac{1}{1-2(n(1-c)^2+\varepsilon)^2}\right\} \ln\left(\frac{1}{\delta}\right)\right)$$

copies of the quantum example state $|\psi_a\rangle$, with probability $\geq 1-\delta$, outputs the string a, without prior knowledge of the underlying distribution D_μ.

Note, however, that the learning algorithm does need to obtain the c-boundedness parameter c as input in advance, but this (in general) does not fix the underlying distribution. Observe also that—since Lemma 10 remains valid for noisy quantum examples—, even though we do not explicitly formulate the result of this subsection for noisy quantum training data, such a generalization is possible by combining the strategies presented in this and the previous subsections.

B Proofs

Proof of Lemma 2 We directly compute the state produced by the algorithm before the measurement is performed:

$$\begin{aligned}(H_\mu \otimes H)|\psi_f\rangle &= \sum_{x\in\{-1,1\}^n}\sum_{j\in\{0,1\}^n}\frac{1}{\sqrt{2}}D_\mu(x)\phi_{\mu,j}(x)\left(|j,0\rangle + (-1)^{f(x)}|j,1\rangle\right)\\ &= \frac{1}{\sqrt{2}}\sum_{j\in\{0,1\}^n}\underbrace{\mathbb{E}_{D_\mu}[\phi_{\mu,j}]}_{=\delta_{j,0\ldots0}}|j,0\rangle + \underbrace{\mathbb{E}_{D_\mu}[g\phi_{\mu,j}]}_{=\hat{g}_\mu(j)}|j,1\rangle.\end{aligned}$$

Hence, the computational basis measurement from step 3 of Algorithm 1 on the last qubit returns 1 with probability $\frac{1}{2}$ and if that is the case, the computational basis measurement on the first n qubits will return j with probability $\left(\hat{g}_\mu(j)\right)^2$, as claimed. □

Proof of Lemma 6 The proof is by direct computation using the Fourier expansion:

$$\begin{aligned}(HGH^{-1})(a,b) &= \frac{1}{2^n}\sum_{c,d\in\{0,1\}^n}(-1)^{c\cdot a+d\cdot b}g(c+d)\\ &= \frac{1}{2^n}\sum_{c,d,j\in\{0,1\}^n}(-1)^{c\cdot a+d\cdot b+j\cdot(c+d)}\hat{g}(j)\\ &= \frac{1}{2^n}\sum_{j\in\{0,1\}^n}\hat{g}(j)\underbrace{\sum_{c\in\{0,1\}^n}(-1)^{c\cdot(a+j)}}_{=2^n\delta_{a,j}}\underbrace{\sum_{d\in\{0,1\}^n}(-1)^{d\cdot(b+j)}}_{=2^n\delta_{b,j}}\\ &= 2^n\hat{g}(a)\delta_{a,b}.\end{aligned}$$

Unitarity of H can be checked easily by exploiting the same identity as in the second to last line of the previous computation. □

Proof of Corollary 1 Using Lemma 6 we can directly compute the diagonal entries of the matrix root and obtain

$$\sqrt{G}(a,a) = \left(H^{-1}\cdot\operatorname{diag}\left(\left\{\sqrt{2^n\hat{g}(j)}\mid j\in\{0,1\}^n\right\}\right)\cdot H\right)(a,a)$$

$$= \frac{1}{2^n} \sum_{j,k \in \{0,1\}^n} (-1)^{c \cdot j + d \cdot k} \sqrt{2^n \hat{g}(j)} \delta_{j,k}$$
$$= \frac{1}{\sqrt{2^n}} \sum_{j \in \{0,1\}^n} \sqrt{\hat{g}(j)}$$

for every $a \in \{0,1\}^n$. □

Proof of Lemma 7 As in the proof of Lemma 4, due to the product structure of all the relevant objects (here our assumption on the form of the noise enters), it suffices to consider the case $n = 1$ in detail. In this case, we have $f^{(a)}(x) = a\tilde{x}$, $g^{(a)}(x) = (-1)^{a\tilde{x}+\xi_x}$ for $\tilde{x} = \frac{1-x}{2}$, $\phi_{\mu,0}(x) = 1$, and $\phi_{\mu,1}(x) = \frac{x-\mu}{\sqrt{1-\mu^2}}$. (We leave out unnecessary indices to improve readability.) We compute

$$\begin{aligned}\hat{g}_\mu^{(a)}(j) &= \mathbb{E}_{D_\mu}[(-1)^{a\tilde{x}+\xi_x} \phi_{\mu,j}(x)] \\ &= \frac{1+\mu}{2} \cdot (-1)^{\xi_1} \cdot \phi_{\mu,j}(1) + \frac{1-\mu}{2} \cdot (-1)^{a+\xi_{-1}} \cdot \phi_{\mu,j}(-1).\end{aligned}$$

By plugging in we now obtain

$$\begin{aligned}\hat{g}_\mu^{(0)}(0) &= \frac{1+\mu}{2} \cdot (-1)^{\xi_1} \cdot 1 + \frac{1-\mu}{2} \cdot (-1)^{\xi_{-1}} \cdot 1, \\ \hat{g}_\mu^{(0)}(1) &= \frac{1+\mu}{2} \cdot (-1)^{\xi_1} \cdot \frac{1-\mu}{\sqrt{1-\mu^2}} + \frac{1-\mu}{2} \cdot (-1)^{\xi_{-1}} \cdot \frac{-1-\mu}{\sqrt{1-\mu^2}}, \\ \hat{g}_\mu^{(1)}(0) &= \frac{1+\mu}{2} \cdot (-1)^{\xi_1} \cdot 1 + \frac{1-\mu}{2} \cdot (-1)^{1+\xi_{-1}} \cdot 1, \\ \hat{g}_\mu^{(1)}(1) &= \frac{1+\mu}{2} \cdot (-1)^{\xi_1} \cdot \frac{1-\mu}{\sqrt{1-\mu^2}} + \frac{1-\mu}{2} \cdot (-1)^{1+\xi_{-1}} \cdot \frac{-1-\mu}{\sqrt{1-\mu^2}}.\end{aligned}$$

So with probability $(\eta^1)^2 + (1-\eta^1)^2 = 1 - 2\eta^1(1-\eta^1)$, namely if $\xi_1 = \xi_{-1} = b \in \{0,1\}$, we obtain

$$\hat{g}_\mu^{(0)}(0) = (-1)^b, \quad \hat{g}_\mu^{(0)}(1) = 0, \quad \hat{g}_\mu^{(1)}(0) = (-1)^b \mu, \quad \hat{g}_\mu^{(1)}(1) = (-1)^b \sqrt{1-\mu^2},$$

and with probability $2\eta^1(1-\eta^1)$, namely if $\xi_1 = b \neq \xi_{-1}$, we obtain

$$\hat{g}_\mu^{(0)}(0) = (-1)^b \mu, \quad \hat{g}_\mu^{(0)}(1) = (-1)^b \sqrt{1-\mu^2}, \quad \hat{g}_\mu^{(1)}(0) = (-1)^b, \quad \hat{g}_\mu^{(1)}(1) = 0.$$

Therefore we obtain: With probability $1-2\eta^1(1-\eta^1)$ the μ-biased Fourier coefficients satisfy

$$\hat{g}_\mu^{(a)}(j) = \begin{cases} (-1)^b(1-j), & \text{for } a = 0 \\ (-1)^b((1-j)\mu + j\sqrt{1-\mu^2}) & \text{for } a = 1 \end{cases},$$

and with probability $2\eta^1(1-\eta^1)$ the μ-biased Fourier coefficients satisfy

$$\hat{g}_\mu^{(a)}(j) = \begin{cases} (-1)^b((1-j)\mu + j\sqrt{1-\mu^2}) & \text{for } a = 0 \\ (-1)^b(1-j), & \text{for } a = 1 \end{cases},$$

which is exactly the claim for $n = 1$. □

Proof of Theorem 8 We want to prove that $\mathbb{P}[\text{Algorithm 6 does not output } a] \leq \delta$, where the probability is w.r.t. both the internal randomness of the algorithm and the random variables.

First observe that, due to (i) in Theorem 7, exactly the same reasoning as in the proof of Theorem 4 shows that the probability of observing $j_{n+1} = 1$ in at most $k-1$ of the m runs of Algorithm 5 (assuming $k \leq \frac{m}{2}$) is bounded by

$$\mathbb{P}\left[\text{Bin}\left(m, \frac{1}{2}\right) \geq m - k\right] \leq \exp\left(-\frac{2\left(\frac{m}{2} - k\right)^2}{m}\right). \tag{B.1}$$

We will now search for the number of observations of $j_{n+1} = 1$ which is required to guarantee that the majority string is correct with high probability. Suppose we observe $j_{n+1} = 1$ in k runs of Algorithm 5, $k \in 2\mathbb{N}$. Again we see that

$$\begin{aligned}\mathbb{P}[\exists 1 \leq i \leq n : a_i \neq o_i] \leq\ & \mathbb{P}[\exists 1 \leq i \leq n : a_i = 0 \neq o_i] \\ & + \mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq o_i].\end{aligned}$$

As "false 1's" can only appear in the case where our noise variables have an influence (compare Theorem 7), we will first find a lower bound on k which guarantees that the probability of the noise variable influence becoming relevant for at least $\frac{k}{5}$ runs is $\leq \frac{\delta}{4}$. Namely, we bound (again via Hoeffding)

$$\begin{aligned}\mathbb{P}\left[\text{Bin}(k, n\rho) \geq \frac{k}{5}\right] &= \mathbb{P}\left[\text{Bin}(k, n\rho) - kn\rho \geq k\left(\frac{1}{5} - n\rho\right)\right] \\ &\leq \exp\left(-2k\left(\frac{1-5n\rho}{5}\right)^2\right).\end{aligned}$$

We now set this last expression $\leq \frac{\delta}{4}$ and rearrange the inequality to

$$k \geq \frac{25}{2(1-5n\rho)^2} \ln\left(\frac{4}{\delta}\right).$$

Now we will find a lower bound on k which guarantees that, if the noise variable influence is relevant in at most $\frac{k}{5}$ of the runs, among the remaining $\frac{4k}{5}$ runs with probability $\geq 1 - \frac{\delta}{4}$ we make at most $\frac{k}{5}$ "false 0" observations. To this end, we bound

(again via Hoeffding)

$$\begin{aligned}
&\mathbb{P}\left[\mathrm{Bin}\left(\frac{4k}{5}, n(1-c)^2\right) \geq \frac{k}{5}\right] \\
&\quad = \mathbb{P}\left[\mathrm{Bin}\left(\frac{4k}{5}, n(1-c)^2\right) - \frac{4kn(1-c)^2}{5} \geq \frac{k}{5} - \frac{4kn(1-c)^2}{5}\right] \\
&\quad \leq \exp\left(-2k\left(\frac{1}{5} - \frac{4n(1-c)^2}{5}\right)^2\right).
\end{aligned}$$

We now set this last expression $\leq \frac{\delta}{4}$ and rearrange the inequality to

$$k \geq \frac{25}{2(1-4n(1-c)^2)^2} \ln\left(\frac{4}{\delta}\right).$$

Hence, by the union bound a sufficient condition for $\mathbb{P}[\exists 1 \leq i \leq n : a_i \neq o_i] \leq \frac{\delta}{2}$ to hold is given by

$$k \geq \frac{25}{2} \max\left\{\frac{1}{(1-5n\rho)^2}, \frac{1}{(1-4n(1-c)^2)^2}\right\} \ln\left(\frac{4}{\delta}\right). \tag{B.2}$$

Combining Eqs. (B.2) and (B.1) we now require

$$\exp\left(-\frac{2\left(\frac{25}{2}\max\left\{\frac{1}{(1-5n\rho)^2}, \frac{1}{(1-4n(1-c)^2)^2}\right\}\ln\left(\frac{4}{\delta}\right) - \frac{m}{2}\right)^2}{m}\right) \overset{!}{\leq} \frac{\delta}{4}.$$

Rearranging gives the sufficient condition

$$m \geq 25 \max\left\{\frac{1}{(1-5n\rho)^2}, \frac{1}{(1-4n(1-c)^2)^2}\right\} \ln\left(\frac{4}{\delta}\right).$$

This proves the claim of the theorem thanks to the union bound. □

Proof of Corollary 2 According to the Lemma 9 it holds that

$$\begin{aligned}
&\left\|H_\mu - H_{\tilde{\mu}}\right\| \\
&\quad \leq \sum_{i=1}^{n} \left\|\mathbb{1} \otimes \ldots \otimes \mathbb{1} \otimes H_{\mu_i} \otimes \mathbb{1} \otimes \ldots \otimes \mathbb{1} - \mathbb{1} \otimes \ldots \otimes \mathbb{1} \otimes H_{\tilde{\mu}_i} \otimes \mathbb{1} \otimes \ldots \otimes \mathbb{1}\right\| \\
&\quad = \sum_{i=1}^{n} \left\|H_{\mu_i} - H_{\tilde{\mu}_i}\right\|.
\end{aligned}$$

Thus it suffices to bound the operator norm of the difference of the 1-qubit biased quantum Fourier transforms. So let $|\varphi\rangle = \sum_{x\in\{-1,1\}} \alpha_x |x\rangle$ be a qubit state. Then

$$(H_{\mu_j} - H_{\tilde{\mu}_j})|\varphi\rangle = \sum_{x\in\{-1,1\}} \sum_{j\in\{0,1\}} \left(\sqrt{D_{\mu_i}(x)}\phi_{\mu_i,j}(x) - \sqrt{D_{\tilde{\mu}_i}(x)}\phi_{\tilde{\mu}_i,j}(x)\right)\alpha_x|j\rangle.$$

We have to bound the (Euclidean) norm of this vector. To achieve this, we will bound (for arbitrary $x \in \{-1, 1\}$ and $j \in \{0, 1\}$) the expression

$$\left|\sqrt{D_{\mu_i}(x)}\phi_{\mu_i,j}(x) - \sqrt{D_{\tilde{\mu}_i}(x)}\phi_{\tilde{\mu}_i,j}(x)\right|^2.$$

This is done by direct computation using $1 - \mu_i^2 \geq 1 - (1-c)^2 \geq c^2$, $1 - \tilde{\mu}_i^2 \geq c^2$ and $|\mu_i - \tilde{\mu}_i| \leq \varepsilon$ as follows:

$$\begin{aligned}
&\left|\sqrt{D_{\mu_i}(x)}\phi_{\mu_i,j}(x) - \sqrt{D_{\tilde{\mu}_i}(x)}\phi_{\tilde{\mu}_i,j}(x)\right| \\
&= \left|\frac{(x_i-\mu_i)\sqrt{1-\tilde{\mu}_i^2}\sqrt{D_{\mu_i}(x)} - (x_i-\tilde{\mu}_i)\sqrt{1-\mu_i^2}\sqrt{D_{\tilde{\mu}_i}(x)}}{\sqrt{1-\tilde{\mu}_i^2}\sqrt{1-\mu_i^2}}\right| \\
&\leq \frac{1}{c^2}\left|(x_i-\mu_i)\sqrt{1-\tilde{\mu}_i^2}\sqrt{D_{\mu_i}(x)} - (x_i-\tilde{\mu}_i)\sqrt{1-\mu_i^2}\sqrt{D_{\tilde{\mu}_i}(x)}\right| \\
&= \frac{1}{c^2}\left|(x_i-\mu_i)\left(\sqrt{1-\tilde{\mu}_i^2}\sqrt{D_{\mu_i}(x)} - \sqrt{1-\mu_i^2}\sqrt{D_{\tilde{\mu}_i}(x)}\right)\right. \\
&\quad \left. + (\tilde{\mu}_i-\mu_i)\sqrt{1-\mu_i^2}\sqrt{D_{\tilde{\mu}_i}(x)}\right| \\
&\leq \frac{1}{c^2}\left(\left|(x_i-\mu_i)\left(\sqrt{1-\tilde{\mu}_i^2}\sqrt{D_{\mu_i}(x)} - \sqrt{1-\mu_i^2}\sqrt{D_{\tilde{\mu}_i}(x)}\right)\right|\right. \\
&\quad \left. + \left|(\tilde{\mu}_i-\mu_i)\sqrt{1-\mu_i^2}\sqrt{D_{\tilde{\mu}_i}(x)}\right|\right) \\
&\leq \frac{1}{c^2}\left((2-c)\left|\sqrt{1-\tilde{\mu}_i^2}\sqrt{D_{\mu_i}(x)} - \sqrt{1-\mu_i^2}\sqrt{D_{\tilde{\mu}_i}(x)}\right| + \varepsilon\right) \\
&\leq \frac{1}{c^2}\left((2-c)\left(\left|\sqrt{D_{\mu_i}(x)} - \sqrt{D_{\tilde{\mu}_i}(x)}\right| + \left|\sqrt{1-\mu_i^2} - \sqrt{1-\tilde{\mu}_i^2}\right|\right) + \varepsilon\right).
\end{aligned}$$

Now note that

$$\begin{aligned}
\left|\left(\sqrt{D_{\mu_i}(x)} - \sqrt{D_{\tilde{\mu}_i}(x)}\right)\left(\sqrt{D_{\mu_i}(x)} + \sqrt{D_{\tilde{\mu}_i}(x)}\right)\right| &= \left|D_{\mu_i}(x) - D_{\tilde{\mu}_i}(x)\right| \\
&= \left|\frac{1+\tilde{x}_i\mu_i}{2} - \frac{1+\tilde{x}_i\tilde{\mu}_i}{2}\right| \\
&= \frac{1}{2}|\mu_i - \tilde{\mu}_i|,
\end{aligned}$$

which implies

$$\begin{aligned}\left|\sqrt{D_{\mu_i}(x)}-\sqrt{D_{\tilde{\mu}_i}(x)}\right| &= \left|\frac{\mu_i-\tilde{\mu}_i}{2\left(\sqrt{D_{\mu_i}(x)}+\sqrt{D_{\tilde{\mu}_i}(x)}\right)}\right| \\ &\leq \frac{\varepsilon}{2}\frac{1}{2\sqrt{\frac{c}{2}}} \\ &= \frac{\varepsilon}{2\sqrt{2c}},\end{aligned}$$

and that moreover

$$\begin{aligned}\left|\left(\sqrt{1-\mu_i^2}-\sqrt{1-\tilde{\mu}_i^2}\right)\left(\sqrt{1-\mu_i^2}+\sqrt{1-\tilde{\mu}_i^2}\right)\right| &= \left|1-\mu_i^2-(1-\tilde{\mu}_i^2)\right| \\ &= \left|\mu_i^2-\tilde{\mu}_i^2\right|,\end{aligned}$$

which in turn implies

$$\begin{aligned}\left|\sqrt{1-\mu_i^2}-\sqrt{1-\tilde{\mu}_i^2}\right| &= \left|\frac{\mu_i^2-\tilde{\mu}_i^2}{\sqrt{1-\mu_i^2}+\sqrt{1-\tilde{\mu}_i^2}}\right| \\ &\leq \frac{|\mu_i+\tilde{\mu}_i|\cdot|\mu_i-\tilde{\mu}_i|}{2\sqrt{1-(1-c)^2}} \\ &\leq \frac{2\varepsilon}{2\sqrt{2c-c^2}} \\ &\leq \frac{\varepsilon}{\sqrt{2}c}.\end{aligned}$$

Hence, we obtain

$$\left|\sqrt{D_{\mu_i}(x)}\phi_{\mu_i,j}(x)-\sqrt{D_{\tilde{\mu}_i}(x)}\phi_{\tilde{\mu}_i,j}(x)\right| \leq \frac{1}{c^2}\left((2-c)\left(\frac{\varepsilon}{2\sqrt{2c}}+\frac{\varepsilon}{\sqrt{2}c}\right)+\varepsilon\right) \leq \gamma\varepsilon,$$

where we defined $\gamma := \frac{1}{c^2}\left((2-c)\frac{3}{2\sqrt{2}c}+1\right)$. This now implies

$$\begin{aligned}\left\|(H_{\mu_j}-H_{\tilde{\mu}_j})|\varphi\rangle\right\|_2 &\leq \sum_{x\in\{-1,1\}}\sum_{j\in\{0,1\}}\left\|\left(\sqrt{D_{\mu_i}(x)}\phi_{\mu_i,j}(x)-\sqrt{D_{\tilde{\mu}_i}(x)}\phi_{\tilde{\mu}_i,j}(x)\right)\alpha_x|j\rangle\right\|_2 \\ &\leq \gamma\varepsilon\sum_{x\in\{-1,1\}}\sum_{j\in\{0,1\}}|\alpha_x| \\ &= 2\gamma\varepsilon\sum_{x\in\{-1,1\}}|\alpha_x| \\ &\leq 2\sqrt{2}\gamma\varepsilon.\end{aligned}$$

Finally, we get

$$\left\| H_\mu - H_{\tilde{\mu}} \right\| \leq \sum_{i=1}^{n} \left\| H_{\mu_i} - H_{\tilde{\mu}_i} \right\| \leq 2\sqrt{2} n \gamma \varepsilon,$$

as claimed. □

References

1. Arunachalam, S., Chakraborty, S., Lee, T., Paraashar, M., de Wolf, R.: Two new results about quantum exact learning (2018). arXiv:1810.00481
2. Arunachalam, S., Grilo, A.B., Sundaram, A.: Quantum hardness of learning shallow classical circuits (2019). arXiv:1903.02840
3. Arunachalam, S., de Wolf, R.: Guest column: a survey of quantum learning theory. SIGACT News **48**, (2017). https://doi.org/10.1145/3106700.3106710. https://pure.uva.nl/ws/files/25255496/p41_arunachalam.pdf
4. Arunachalam, S., de Wolf, R.: Optimal quantum sample complexity of learning algorithms. J. Mach. Learn. Res. **19**(71), 1–36 (2018). http://jmlr.org/papers/v19/18-195.html
5. Atıcı, A., Servedio, R.A.: Quantum algorithms for learning and testing juntas. Quantum Inf. Process. **6**(5), 323–348 (2007). https://doi.org/10.1007/s11128-007-0061-6
6. Barnum, H., Knill, E.: Reversing quantum dynamics with near-optimal quantum and classical fidelity (2000). arXiv:quant-ph/0004088
7. Benioff, P.: The computer as a physical system: a microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. J. Stat. Phys. **22**(5), 563–591 (1980). https://doi.org/10.1007/BF01011339
8. Bernstein, E., Vazirani, U.: Quantum complexity theory. In: Kosaraju R. (ed.) Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, pp. 11–20. ACM, New York (1993). https://doi.org/10.1145/167088.167097
9. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM **50**(4), 506–519 (2003). https://doi.org/10.1145/792538.792543
10. Bshouty, N.H., Jackson, J.C.: Learning dnf over the uniform distribution using a quantum example oracle. SIAM J. Comput. **28**(3), 1136–1153 (1998). https://doi.org/10.1137/S0097539795293123
11. Cross, A.W., Smith, G., Smolin, J.A.: Quantum learning robust against noise. Phys. Rev. A **92**(1), 97 (2015). https://doi.org/10.1103/PhysRevA.92.012327
12. Feynman, R.P.: Quantum mechanical computers. Opt. News **11**(2), 11 (1985). https://doi.org/10.1364/ON.11.2.000011
13. Grilo, A.B., Kerenidis, I., Zijlstra, T.: Learning with errors is easy with quantum samples (2017). Phys. Rev. A **99**(3), 032314 (2019). https://doi.org/10.1103/PhysRevA.99.032314
14. Hausladen, P., Wootters, W.K.: A 'pretty good' measurement for distinguishing quantum states. J. Mod. Opt. **41**(12), 2385–2390 (1994). https://doi.org/10.1080/09500349414552221
15. Hoeffding, W.: Probability inequalities for sums of bounded random variables. J. Am. Stat. Assoc. **58**(301), 13–30 (1963). https://doi.org/10.1080/01621459.1963.10500830
16. Ivanyos, G., Prakash, A., Santha, M. (eds.): On Learning Linear Functions from Subset and Its Applications in Quantum Computing. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern (2018). https://doi.org/10.4230/LIPICS.ESA.2018.66
17. Kanade, V., Rocchetto, A., Severini, S.: Learning dnfs under product distributions via μ-biased quantum fourier sampling. Quantum Inf. Comput. **19**(15&16), 1261–1278 (2019)
18. Lyubashevsky, V.: The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In: Chekuri C. (ed.) Approximation, Randomization and Combinatorial optimization. Lecture Notes in Computer Science, vol. 3624, pp. 378–389. Springer, Berlin (2005). https://doi.org/10.1007/11538462_32
19. Montanaro, A.: The quantum query complexity of learning multilinear polynomials. Inf. Process. Lett. **112**(11), 438–442 (2012). https://doi.org/10.1016/j.ipl.2012.03.002

20. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information, 10 Anniversary edn. Cambridge University Press, Cambridge (2010)
21. O'Donnell, R.: Analysis of Boolean Functions. Cambridge University Press, Cambridge (2014)
22. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 1–40 (2009). https://doi.org/10.1145/1568318.1568324
23. Ristè, D., da Silva, M.P., Ryan, C.A., Cross, A.W., Córcoles, A.D., Smolin, J.A., Gambetta, J.M., Chow, J.M., Johnson, B.R.: Demonstration of quantum advantage in machine learning. npj Quantum Inf. **3**(1), 16 (2017). https://doi.org/10.1038/s41534-017-0017-3
24. Servedio, R.A., Gortler, S.J.: Equivalences and separations between quantum and classical learnability. SIAM J. Comput. **33**(5), 1067–1092 (2004). https://doi.org/10.1137/S0097539704412910
25. Shalev-Shwartz, S., Ben-David, S.: Understanding machine learning: from theory to algorithms. Cambridge University Press, Cambridge (2014)
26. Valiant, L.G.: A theory of the learnable. Commun. ACM **27**(11), 1134–1142 (1984). https://doi.org/10.1145/1968.1972
27. Vershynin, R.: High-Dimensional Probability: An Introduction with Applications in Data Science. Cambridge Series in Statistical and Probabilistic Mathematics, vol. 47. Cambridge University Press, Cambridge (2018)

Affiliations

Matthias C. Caro[1]

[1] Department of Mathematics, Technische Universität München, Boltzmannstrasse 3, 85748 Garching, Germany