MIT Open Access Articles

Quantum Algorithm for Data Fitting

P H Y S I C A L   R E V I E W   L E T T E R S

℘

Quantum Algorithm for Data Fitting

Nathan Wiebe,[1] Daniel Braun,[2,3] and Seth Lloyd[4]

[1]*Institute for Quantum Computing and Department of Combinatorics and Optimization, University of Waterloo,
200 University Ave., West, Waterloo, Ontario, Canada*
[2]*Laboratoire de Physique Théorique, Université Paul Sabatier, 118, Route de Narbonne, F-31062 Toulouse, France*
[3]*CNRS, LPT (IRSAMC), F-31062 Toulouse, France*
[4]*Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

We provide a new quantum algorithm that efficiently determines the quality of a least-squares fit over an exponentially large data set by building upon an algorithm for solving systems of linear equations efficiently [Harrow *et al.*, Phys. Rev. Lett. **103**, 150502 (2009)]. In many cases, our algorithm can also efficiently find a concise function that approximates the data to be fitted and bound the approximation error. In cases where the input data are pure quantum states, the algorithm can be used to provide an efficient parametric estimation of the quantum state and therefore can be applied as an alternative to full quantum-state tomography given a fault tolerant quantum computer.

Invented as early as 1794 by Carl Friedrich Gauss, fitting data to theoretical models has become over the centuries one of the most important tools in all of quantitative science [1]. Typically, a theoretical model depends on a number of parameters, and leads to functional relations between data that will depend on those parameters. Fitting a large amount of experimental data to the functional relations allows one to obtain reliable estimates of the parameters. If the amount of data becomes very large, fitting can become very costly. Examples include inversion problems of x-ray or neutron scattering data for structure analysis, or high-energy physics with gigabytes of data produced per second at the LHC. Typically, structure analysis starts from a first guess of the structure, and then iteratively tries to improve the fit to the experimental data by testing variations of the structure. It is therefore often desirable to test many *different models*, and compare the best possible fits they provide before committing to one for which one extracts then the parameters from the fit. Obtaining a good fit with a relatively small number of parameters compared to the amount of data can be considered a form of data compression. Indeed, also for numerically calculated data, such as many-body wave-functions in molecular engineering, efficient fitting of the wave-functions to simpler models would be highly desirable.

With the rise of quantum information theory, one might wonder if a quantum algorithm can be found that solves these problems efficiently. The discovery that exploiting quantum mechanical effects might lead to enhanced computational power compared to classical information processing has triggered large-scale research aimed at finding quantum algorithms which are more efficient than the best classical counterparts [2–7]. Although fault-tolerant quantum computation remains out of reach at present, quantum simulation is already now on the verge of providing answers to questions concerning the states of complex systems that

are beyond classical computability [8,9]. Recently, a quantum algorithm (called HHL in the following) was introduced that efficiently solves a linear equation, $\mathbf{F}\mathbf{x} = \mathbf{b}$, with given vector $\mathbf{b}$ of dimension N and sparse Hermitian matrix $\mathbf{F}$ [10]. "Efficient solution" means that the expectation value $\langle \mathbf{x} | \mathbf{M} | \mathbf{x} \rangle$ of an arbitrary poly-size Hermitian operator $\mathbf{M}$ can be found in roughly $\mathcal{O}(s^4 \kappa^2 \log(N)/\epsilon)$ steps [11], where κ is the condition number of $\mathbf{F}$, i.e., the ratio between the largest and smallest eigenvalue of $\mathbf{F}$, s denotes the sparseness (i.e., the maximum number of nonzero matrix elements of $\mathbf{F}$ in any given row or column), and ϵ is the maximum allowed distance between the $|x\rangle$ found by the computer and the exact solution. In contrast, they show that it is unlikely that classical computers can efficiently solve similar problems because it would imply that quantum computers are no more powerful than classical computers.

While it has remained unclear so far whether expectation values of the form $\langle \mathbf{x} | \mathbf{M} | \mathbf{x} \rangle$ provide answers to computationally important questions, we provide here an adaption of the algorithm to the problem of data fitting that allows one to efficiently obtain the quality of a fit without having to learn the fit-parameters. Our algorithm is particularly useful for fitting data *efficiently computed* by a quantum computer or quantum simulator, especially if an evolution can be efficiently simulated but no known method exists to efficiently learn the resultant state. For example, our algorithm could be used to efficiently find a concise matrix–product state approximation to a ground state yielded by a quantum many–body simulator and assess the approximation error. More complicated states can be used in the fit if the quantum computer can efficiently prepare them. Fitting quantum states to a set of known functions is an interesting alternative to performing full quantum-state tomography [13].

Least-squares fitting.—The goal in least-squares fitting is to find a simple continuous function that well approximates

a discrete set of N points $\{x_i, y_i\}$. The function is constrained to be linear in the fit parameters $\boldsymbol{\lambda} \in \mathbb{C}^M$, but it can be nonlinear in $\mathbf{x}$. For simplicity we consider $x \in \mathbb{C}$, but the generalization to higher dimensional x is straightforward. Our fit function is then of the form

$$f(x, \boldsymbol{\lambda}) := \sum_{j=1}^{M} f_j(x)\lambda_j$$

where λ_j is a component of $\boldsymbol{\lambda}$ and $f(x, \boldsymbol{\lambda})$: $\mathbb{C}^{M+1} \mapsto \mathbb{C}$. The optimal fit parameters can be found by minimizing

$$E = \sum_{i=1}^{N} |f(x_i, \boldsymbol{\lambda}) - y_i|^2 = |\mathbf{F}\boldsymbol{\lambda} - \mathbf{y}|^2 \qquad (1)$$

over all $\boldsymbol{\lambda}$, where we have defined the $N \times M$ matrix $\mathbf{F}$ through $\mathbf{F}_{ij} = f_j(x_i)$, $\mathbf{F}^t$ is its transpose, and $\mathbf{y}$ denotes the column vector $(y_1, \ldots, y_N)^t$. Also, following HHL, we assume without loss of generality that $\frac{1}{\kappa^2} \leq \|\mathbf{F}^\dagger \mathbf{F}\| \leq 1$ and $\frac{1}{\kappa^2} \leq \|\mathbf{F}\mathbf{F}^\dagger\| \leq 1$ [10]. Throughout this Letter we use $\|\cdot\|$ to denote the spectral norm.

Given that $\mathbf{F}^\dagger \mathbf{F}$ is invertible, the fit parameters that give the least square error are found by applying the Moore-Penrose pseudoinverse [14] of $\mathbf{F}$, $\mathbf{F}^+$, to $\mathbf{y}$:

$$\boldsymbol{\lambda} = \mathbf{F}^+ \mathbf{y} = (\mathbf{F}^\dagger \mathbf{F})^{-1} \mathbf{F}^\dagger \mathbf{y}. \qquad (2)$$

A proof that (2) gives an optimal $\boldsymbol{\lambda}$ for a least-squares fit is given in the supplemental material [15].

The algorithm consists of three subroutines: a quantum algorithm for performing the pseudoinverse, an algorithm for estimating the fit quality and an algorithm for learning the fit parameters $\boldsymbol{\lambda}$.

1. Fitting algorithm.—Our algorithm uses a quantum computer and oracles that output quantum states that encode the matrix elements of $\mathbf{F}$ to approximately prepare $\mathbf{F}^+ \mathbf{y}$. The matrix multiplications, and inversions, are implemented using an improved version of the HHL algorithm [10] that utilizes recent developments in quantum simulation algorithms.

Input:—A quantum state $|\mathbf{y}\rangle = \sum_{p=M+1}^{M+N} \mathbf{y}_p |p\rangle / |\mathbf{y}|$ that stores the data $\mathbf{y}$, an upper bound (denoted κ) for the square roots of the condition numbers of $\mathbf{F}\mathbf{F}^\dagger$ and $\mathbf{F}^\dagger \mathbf{F}$, the sparseness of $\mathbf{F}$ (denoted s) and an error tolerance ϵ.

Output:—A quantum state $|\boldsymbol{\lambda}\rangle$ that is approximately proportional to the optimal fit parameters $\boldsymbol{\lambda}/|\boldsymbol{\lambda}|$ up to error ϵ as measured by the Euclidean norm.

Computational model:—We have a universal quantum computer equipped with oracles that, when queried about a non–zero matrix element in a given row, yield a quantum state that encodes a requested bit of a binary encoding the column number or value of a nonzero matrix element of $\mathbf{F}$ in a manner similar to those in [16]. We also assume a quantum black box is provided that yields copies of the input state $|\mathbf{y}\rangle$ on demand.

Query complexity:—The number of oracle queries used is

$$\tilde{\mathcal{O}}(\log(N)(s^3\kappa^6)/\epsilon), \qquad (3)$$

where $\tilde{\mathcal{O}}$ notation implies an upper bound on the scaling of a function, suppressing all subpolynomial functions. Alternatively, the simulation method of [17,18] can be used to achieve a query complexity of

$$\tilde{\mathcal{O}}(\log(N)(s\kappa^6)/\epsilon^2). \qquad (4)$$

Analysis of algorithm.—The operators $\mathbf{F}$ and $\mathbf{F}^\dagger$ are implemented using an isometry superoperator $\mathbf{I}$ to represent them as Hermitian operators on $\mathbb{C}^{N+M}$. The isometry has the following action on a matrix $\mathbf{X}$:

$$\mathbf{I}: \mathbf{X} \mapsto \begin{pmatrix} 0 & \mathbf{X} \\ \mathbf{X}^\dagger & 0 \end{pmatrix}. \qquad (5)$$

These choices are convenient because $\mathbf{I}(\mathbf{F}^\dagger)|\mathbf{y}\rangle$ contains $\mathbf{F}^\dagger \mathbf{y}/|\mathbf{y}|$ in its first M entries. We also assume for simplicity that $|\mathbf{I}(\mathbf{F}^\dagger)|\mathbf{y}\rangle| = 1$. This can easily be relaxed by dividing $\mathbf{I}(\mathbf{F}^\dagger)|\mathbf{y}\rangle$ by $|\mathbf{F}^\dagger \mathbf{y}|$.

Preparing $\mathbf{I}(\mathbf{F}^\dagger)|\mathbf{y}\rangle$.—The next step is to prepare the state $\mathbf{I}(\mathbf{F}^\dagger)|\mathbf{y}\rangle$. This is not straightforward because $\mathbf{I}(\mathbf{F}^\dagger)$ is a Hermitian, rather than unitary, operator. We implement the Hermitian operator using the same phase estimation trick that HHL use to enact the inverse of a Hermitian operator, but instead of dividing by the eigenvalues of each eigenstate we multiply each eigenstate by its eigenvalue. We describe the relevant steps below. For more details, see [10].

The algorithm first prepares an ancilla state for a large integer T that is of order N

$$|\Psi_0\rangle = \sqrt{\frac{2}{T}} \sum_{\tau=0}^{T-1} \sin\left(\frac{\pi(\tau + 1/2)}{T}\right) |\tau\rangle \otimes |\mathbf{y}\rangle. \qquad (6)$$

It then maps $|\Psi_0\rangle$ to

$$\sqrt{\frac{2}{T}} \sum_{\tau=0}^{T-1} \sin\left(\frac{\pi(\tau + 1/2)}{T}\right) |\tau\rangle \otimes e^{-i\mathbf{I}(\mathbf{F}^\dagger)\tau t_0/T}|\mathbf{y}\rangle, \qquad (7)$$

for $t_0 \in \mathcal{O}(\kappa/\epsilon)$. We know from work on quantum simulation that $\exp(-i\mathbf{I}(\mathbf{F}^\dagger)\tau t_0/T)$ can be implemented within error $\mathcal{O}(\epsilon)$ in the 2-norm using $\tilde{\mathcal{O}}(\log(N)s^3 t_0/T)$ quantum operations, if $\mathbf{F}$ has sparseness s [19]. Alternatively, the method of [17,18] gives query complexity $\tilde{\mathcal{O}}(\log(N)s\tau t_0/(\epsilon T))$. If we write $|\mathbf{y}\rangle = \sum_{j=1}^{N} \beta_j |\mu_j\rangle$, where $|\mu_j\rangle$ are the eigenvectors of $\mathbf{I}(\mathbf{F}^\dagger)$ with eigenvalue E_j we obtain

$$\sqrt{\frac{2}{T}} \sum_{\tau=0}^{T-1} \sin\left(\frac{\pi(\tau + 1/2)}{T}\right) e^{-iE_j\tau t_0/T}|\tau\rangle \otimes \beta_j |\mu_j\rangle. \qquad (8)$$

The quantum Fourier transform is then applied to the first register and, after labeling the Fourier coefficients $\alpha_{k|j}$, the state becomes

$$\sum_{j=1}^{N} \sum_{k=0}^{T-1} \alpha_{k|j} \beta_j |k\rangle |\mu_j\rangle. \tag{9}$$

HHL show that the Fourier coefficients are small unless the eigenvalue $E_j \approx \tilde{E}_k := 2\pi k/t_0$, and $t_0 \in \mathcal{O}(\kappa/\epsilon)$ is needed to ensure that the error from approximating the eigenvalue is at most ϵ. It can be seen using the analysis in [10] that after relabeling $|k\rangle$ as $|\tilde{E}_k\rangle$, and taking $T \in \mathcal{O}(N)$, (9) is exponentially close to $\sum_{j=1}^{N} \beta_j |\tilde{E}_j\rangle |\mu_j\rangle$.

The final step is to introduce an ancilla system and perform a controlled unitary on it that rotates the ancilla state from $|0\rangle$ to $\sqrt{1 - C^2 \tilde{E}_j^2}|0\rangle + C\tilde{E}_j|1\rangle$, where $C \in \mathcal{O}(\max_j |E_j|)^{-1}$ because the state would not be properly normalized if C were larger. The probability of measuring the ancilla to be 1 is $\mathcal{O}(1/\kappa^2)$ since CE_j is at least $\mathcal{O}(1/\kappa)$. $\mathcal{O}(\kappa^2)$ repetitions are therefore needed to guarantee success with high probability, and amplitude amplification can be used to reduce the number of repetitions to $\mathcal{O}(\kappa)$ [10]. HHL show that either $\mathcal{O}(1/\kappa^2)$ or $\mathcal{O}(1/\kappa)$ attempts are also needed to successfully perform $\mathbf{I}(\mathbf{F})^{-1}$ depending on whether amplitude amplification is used.

The cost of implementing $\mathbf{I}(\mathbf{F}^\dagger)$ is the product of the cost of simulating $\mathbf{I}(\mathbf{F}^\dagger)$ for time κ/ϵ and the number of repetitions required to obtain a successful result, which scales as $O(\kappa)$. The improved simulation method of Childs and Kothari [19] allows the simulation to be performed in time $\tilde{\mathcal{O}}(\log(N)s^3\kappa/\epsilon)$, where s is the sparseness of $\mathbf{F}$; therefore, $\mathbf{I}(\mathbf{F}^\dagger)|\mathbf{y}\rangle$ can be prepared using $\tilde{\mathcal{O}}(\log(N)s^3\kappa^2/\epsilon)$ oracle calls. The cost of performing the inversion using the simulation method of [17,18] is found by substituting $s \rightarrow s^{1/3}/\epsilon$ into this or any of our subsequent results.

Inverting $\mathbf{F}^\dagger \mathbf{F}$.—We then finish the algorithm by applying $(\mathbf{F}^\dagger \mathbf{F})^{-1}$ using the method of HHL [10]. Note that the existence of $(\mathbf{F}^\dagger \mathbf{F})^{-1}$ is implied by a well-defined fitting-problem, in the sense that a zero eigenvalue of $\mathbf{F}^\dagger \mathbf{F}$ would result in a degenerate direction of the quadratic form (1). The operator $\mathbf{F}^\dagger \mathbf{F} \in \mathbb{C}^{M \times M}$ is Hermitian and hence amenable to the linear systems algorithm. We do, however, need to extend the domain of the operator to make it compatible with $|\mathbf{y}\rangle$ which is in a Hilbert space of dimension $N + M$. We introduce $\mathbf{A}$ to denote the corresponding operator,

$$\mathbf{A} := \begin{pmatrix} \mathbf{F}^\dagger \mathbf{F} & 0 \\ 0 & \mathbf{F}\mathbf{F}^\dagger \end{pmatrix} = \mathbf{I}(\mathbf{F})^2. \tag{10}$$

If we define $|\boldsymbol{\lambda}\rangle \in \mathbb{C}^{N+M}$ to be a state of the form $|\boldsymbol{\lambda}\rangle = \sum_{j=1}^{M} \lambda_j |j\rangle$ up to a normalizing constant, then $\mathbf{F}^\dagger \mathbf{F}\boldsymbol{\lambda}$ is proportional to $\mathbf{A}|\boldsymbol{\lambda}\rangle$ up to a normalizing constant. This means that we can find a vector that is proportional to the least-squares fit parameters by inversion via

$$|\boldsymbol{\lambda}\rangle = \mathbf{A}^{-1} \mathbf{I}(\mathbf{F}^\dagger)|\mathbf{y}\rangle. \tag{11}$$

This can be further simplified by noting that

$$\mathbf{A}^{-1} = \mathbf{I}(\mathbf{F})^{-2}. \tag{12}$$

Amplitude amplification does not decrease the number of attempts needed to implement $\mathbf{A}^{-1}$ in (11) because the algorithm require reflections about $\mathbf{I}(\mathbf{F}^\dagger)|\mathbf{y}\rangle$, which requires $\mathcal{O}(\kappa)$ repetitions to prepare.

Since amplitude amplification provides no benefit for implementing $\mathbf{A}^{-1}$, $\mathcal{O}(\kappa^5)$ repetitions are needed to implement $\mathbf{A}^{-1}\mathbf{I}(\mathbf{F}^\dagger)$. This is a consequence of the fact that the probability of successfully performing each $\mathbf{I}(\mathbf{F})^{-1}$ is $\mathcal{O}(1/\kappa^2)$ and the probability of performing $\mathbf{I}(\mathbf{F}^\dagger)$ is $\mathcal{O}(1/\kappa)$ (if amplitude amplification is used). The cost of performing the simulations involved in each attempt is $\tilde{\mathcal{O}}(\log(N)s^3\kappa/\epsilon)$ and hence the required number of oracle calls scales as

$$\tilde{\mathcal{O}}(\log(N)(s^3\kappa^6/\epsilon)). \tag{13}$$

Although the algorithm yields $|\boldsymbol{\lambda}\rangle$ efficiently, it may be exponentially expensive to learn $|\boldsymbol{\lambda}\rangle$ via tomography; however, we show below that a quantum computer can assess the quality of the fit efficiently.

2. Estimating fit quality.—We will now show that we can efficiently estimate the fit quality E even if M is exponentially large and without having to determine the fit-parameters. For this problem, note that due to the isometry (5) $E = ||\mathbf{y}\rangle - \mathbf{I}(\mathbf{F})|\boldsymbol{\lambda}\rangle||^2$. We assume the prior computational model. We are also provided a desired error tolerance, ϵ, and wish to determine the quality of the fit within error δ.

Input:—A constant $\delta > 0$ and all inputs required by algorithm 1.

Output:—An estimate of $|\langle \mathbf{y}|\mathbf{I}(\mathbf{F})|\boldsymbol{\lambda}\rangle|^2$ accurate within error δ.

Query complexity:—

$$\tilde{\mathcal{O}}\left(\log(N)\frac{s^3\kappa^4}{\epsilon\delta^2}\right). \tag{14}$$

Algorithm.—We begin by preparing the state $|\mathbf{y}\rangle \otimes |\mathbf{y}\rangle$ using the provided state preparation black box. We then use the prior algorithm to construct the state

$$\mathbf{I}(\mathbf{F})\mathbf{A}^{-1}\mathbf{I}(\mathbf{F}^\dagger)|\mathbf{y}\rangle \otimes |\mathbf{y}\rangle = \mathbf{I}(\mathbf{F})^{-1}\mathbf{I}(\mathbf{F}^\dagger)|\mathbf{y}\rangle \otimes |\mathbf{y}\rangle, \tag{15}$$

within error $\mathcal{O}(\epsilon)$. The cost of implementing $\mathbf{I}(\mathbf{F})^{-1}\mathbf{I}(\mathbf{F}^\dagger)$ (with high probability) within error ϵ is

$$\tilde{\mathcal{O}}\left(\log(N)\frac{s^3\kappa^4}{\epsilon}\right). \tag{16}$$

The swap test [20] is then used to determine the accuracy of the fit. The swap test is a method that can be used to distinguish $|\mathbf{y}\rangle$ and $\mathbf{I}(\mathbf{F})|\boldsymbol{\lambda}\rangle$ by performing a swap operation on the two quantum states controlled by a qubit in the state $(|0\rangle + |1\rangle)/\sqrt{2}$. The Hadamard operation is then applied to the control qubit and the control qubit is then measured in the computational basis. The test concludes that the states are different if the outcome is "1".

The probability of observing an outcome of 1 is $(1 - |\langle \mathbf{y} | \mathbf{I}(\mathbf{F}) | \boldsymbol{\lambda} \rangle|^2)/2$ for our problem.

The overlap between the two quantum states can be learned by statistically sampling the outcomes from many instances of the swap test. The value of $|\langle \mathbf{y} | \mathbf{I}(\mathbf{F}) | \boldsymbol{\lambda} \rangle|^2$ can be approximated using the sample mean of this distribution. It follows from estimates of the standard deviation of the mean that $\mathcal{O}(1/\delta^2)$ samples are required to estimate the mean within error $\mathcal{O}(\delta)$. The cost of algorithm 2 is then found by multiplying (16) by $1/\delta^2$.

The quantity E can be estimated from the output of algorithm 2 by $E \leq 2(1 - |\langle \mathbf{y} | \mathbf{I}(\mathbf{F}) | \boldsymbol{\lambda} \rangle|)$. Taylor series analysis shows that the error in the upper bound for E is also $\mathcal{O}(\delta)$.

There are several important limitations to this technique. First, if $\mathbf{F}$ is not sparse (meaning $s \in \mathcal{O}(\text{poly}(N))$) then the algorithm may not be efficient because the quantum simulation step used in the algorithm may not be efficient. As noted in previous results [16–18], we can generalize our results to systems where $\mathbf{F}$ is nonsparse if there exists a set of efficient unitary transformations U_j such that $\mathbf{I}(\mathbf{F}) = \sum_j U_j H_j U_j^\dagger$ where each H_j is sparse and Hermitian. Also, in many important cases (such as fitting to experimental data) it may not be possible to prepare the initial state $|\mathbf{y}\rangle$ efficiently. For this reason, our algorithm is better suited for approximating the output of quantum devices than the classical outputs of experiments. Finally, algorithm 2 only provides an efficient estimate of the fit quality and does not provide $\boldsymbol{\lambda}$; however, we can use it to determine whether a quantum state has a concise representation within a family of states. If algorithm 2 can be used to find such a representation, then the parameters $|\boldsymbol{\lambda}\rangle$ can be learned via state tomography. We discuss this approach below.

3. Learning $\boldsymbol{\lambda}$.—This method can also be used to find a concise fit function that approximates $\mathbf{y}$. Specifically, we use statistical sampling and quantum-state tomography to find a concise representation for the quantum state using M' parameters. The resulting algorithm is efficient if $M' \in \mathcal{O}(\text{polylog}(N))$.

Input:—As algorithm 2, but in addition with an integer $M' \in \mathcal{O}(\text{polylog}(M))$ that gives the maximum number of fit functions allowed in the fit.

Output:—A classical bit string approximating $|\boldsymbol{\lambda}\rangle$ to precision ϵ, a list of the M' fit functions that comprise $|\boldsymbol{\lambda}\rangle$ and $|\langle \mathbf{y} | \mathbf{I}(\mathbf{F}) | \boldsymbol{\lambda} \rangle|^2$ calculated to precision δ.

Computational model:—As algorithm 1, but the oracles can be controlled to either fit the state to all M fit functions or any subset consisting of M' fit functions.

Query complexity:—

$$\tilde{\mathcal{O}}\left(\log(N) s^3 \left(\frac{\kappa^4}{\epsilon \delta^2} + \frac{M'^2 \kappa^6}{\epsilon^3} \right) \right).$$

Algorithm.—The first step of the algorithm is to prepare the state $|\boldsymbol{\lambda}\rangle$ using algorithm 1. The state is then measured $\mathcal{O}(M')$ times and a histogram of the measurement

outcomes is constructed. Since the probability of measuring each of these outcomes is proportional to their relevance to the fit, we are likely to find M' of the most likely outcomes by sampling the state $\mathcal{O}(M')$ times.

After choosing the M' most significant fit functions, we remove all other fit functions from the fit and prepare the state $|\boldsymbol{\lambda}\rangle$ using the reduced set of fit functions. Compressed sensing [21–23] is then used to reconstruct $|\boldsymbol{\lambda}\rangle$ within $\mathcal{O}(\epsilon)$ error. The idea of compressed sensing is that a low-rank density matrix can be uniquely determined (with high probability) by a small number of randomly chosen measurements. A convex optimization routine is then used to reconstruct the density matrix from the expectation values found for each of the measurements.

Compressed sensing requires $\mathcal{O}(M' \log(M')^2)$ measurement settings to reconstruct pure states, and observation 1 of [21] implies that $\mathcal{O}(M'/\epsilon^2)$ measurements are needed for each setting to ensure that the reconstruction error is $\mathcal{O}(\epsilon)$; therefore, $\mathcal{O}(M'^2 \log(M')^2/\epsilon^2)$ measurements are needed to approximate the state within error $\mathcal{O}(\epsilon)$. The total cost of learning $|\boldsymbol{\lambda}\rangle$ is the number of measurements needed for tomography multiplied by the cost of preparing the state and thus scales as

$$\tilde{\mathcal{O}}\left(\log(N) \frac{s^3 M'^2 \kappa^6}{\epsilon^3} \right), \tag{17}$$

which subsumes the cost of measuring $|\boldsymbol{\lambda}\rangle$ to find the most significant M' fit functions.

Finally, we measure the quality of the fit using algorithm 2. The total cost of estimating $|\boldsymbol{\lambda}\rangle$ and the fit quality is thus the sum of (17) and (16), as claimed.

Remark.—The quality of the resulting fit that is yielded by this algorithm depends strongly on the set of fit functions that are used. If the fit functions are chosen well, fewer than M' fit functions are used to estimate $|\mathbf{y}\rangle$ with high fidelity. Conversely, $\mathcal{O}(N)$ fit functions may be needed to achieve the desired error tolerance if the fit functions are chosen poorly. Fortunately, the efficiency of algorithm 2 allows the user to search many sets of possible fit functions for a concise and accurate model within a large set of potential models.

[1] O. Bretscher, *Linear Algebra With Applications* (Prentice Hall, Upper Saddle River, NJ, 1995), 3rd ed.

[2] P. W. Shor, in *Proc. 35th Annu. Symp. Foundations of Computer Science*, edited by S. Goldwasser (IEEE Computer Society, Los Alamitos, CA, 1994), p. 124

[3] D. Simon (IEEE Computer Society, Los Alamitos, CA, 1994).

[4] L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997).

[5] W. van Dam and S. Hallgren, arXiv:quant-ph/0011067v2.

[6] D. Aharonov, Z. Landau, and J. Makowsky, arXiv:quant-ph/0611156.

[7] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, *Proc. 35th ACM Symposium on Theory of Computing (STOC 2003)* (2002), p. 59.

[8] J. T. Barreiro, M. Müller, P. Schindler, D. Nigg, T. Monz, M. Chwalla, M. Hennrich, C. F. Roos, P. Zoller, and R. Blatt, Nature (London) **470**, 486 (2011).

[9] J. Simon, W. S. Bakr, R. Ma, M. E. Tai, P. M. Preiss, and M. Greiner, Nature (London) **472**, 307 (2011).

[10] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[11] The result in [10] incorrectly cites the results of Theorem 2 of [12] leading to the conclusion that the cost of linear inversion scales with the sparseness as $\tilde{\mathcal{O}}(s^2)$ rather than $\tilde{\mathcal{O}}(s^4)$.

[12] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Commun. Math. Phys. **270**, 359 (2007).

[13] Z. Hradil, Phys. Rev. A **55**, R1561 (1997).

[14] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed. (Springer, New York, 2003).

[15] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.109.050505 for a brief proof that the Moore-Penrose pseudoinverse gives the fit parameters that minimize the least square error when applied to a vector that stores the data to be fit.

[16] N. Wiebe, D. W. Berry, P. Høyer, and B. C. Sanders, J. Phys. A **44**, 445308 (2011).

[17] A. M. Childs, Commun. Math. Phys. **294**, 581 (2009).

[18] D. W. Berry and A. M. Childs, Quantum Inf. Comput. **12**, 29 (2012).

[19] A. Childs and R. Kothari, in *Theory of Quantum Computation, Communication, and Cryptography*, Lecture Notes in Computer Science Vol. 6519, edited by W. van Dam, V. Kendon, and S. Severini (Springer, Berlin, Heidelberg, 2011), p. 94, ISBN 978-3-642-18072-9.

[20] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Phys. Rev. Lett. **87**, 167902 (2001).

[21] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, Phys. Rev. Lett. **105**, 150401 (2010).

[22] A. Shabani, R. L. Kosut, M. Mohseni, H. Rabitz, M. A. Broome, M. P. Almeida, A. Fedrizzi, and A. G. White, Phys. Rev. Lett. **106**, 100401 (2011).

[23] A. Shabani, M. Mohseni, S. Lloyd, R. L. Kosut, and H. Rabitz, Phys. Rev. A **84**, 012107 (2011).