Continuous-variable quantum Gaussian process regression and quantum singular value decomposition of nonsparse low-rank matrices

Siddhartha Das,[1] George Siopsis,[2] and Christian Weedbrook[3]
[1]*Hearne Institute for Theoretical Physics, Louisiana State University, Baton Rouge, Louisiana 70803, USA*
[2]*Department of Physics and Astronomy, The University of Tennessee, Knoxville, Tennessee 37966-1200, USA*
[3]*Xanadu, 372 Richmond St. W, Toronto, M5V 2L7, Canada*

(Received 4 July 2017; published 12 February 2018)

With the significant advancement in quantum computation during the past couple of decades, the exploration of machine-learning subroutines using quantum strategies has become increasingly popular. Gaussian process regression is a widely used technique in supervised classical machine learning. Here we introduce an algorithm for Gaussian process regression using continuous-variable quantum systems that can be realized with technology based on photonic quantum computers under certain assumptions regarding distribution of data and availability of efficient quantum access. Our algorithm shows that by using a continuous-variable quantum computer a dramatic speedup in computing Gaussian process regression can be achieved, i.e., the possibility of exponentially reducing the time to compute. Furthermore, our results also include a continuous-variable quantum-assisted singular value decomposition method of nonsparse low rank matrices and forms an important subroutine in our Gaussian process regression algorithm.

DOI: 10.1103/PhysRevA.97.022315

I. INTRODUCTION

One of the current technological needs in the area of computer science is finding an efficient and faster way of manipulating large data sets, and extracting worthwhile inferences. In the last decade, machine-learning techniques have been used to perform many tasks involving big data. In 1959, Samuel defined machine learning as the "field of study that gives computers the ability to learn without being explicitly programed" [1]. Machine learning has not only helped us better understand the human genome, but has also made self-driving cars, practical speech recognition, effective web search, etc., possible [2,3].

One of the two machine-learning methods involves supervised learning (with the other task being unsupervised learning) [4]. It is the problem of learning input-output mappings from an empirical (training) data set. Depending on the nature of the output, the problem of supervised learning can be categorized under two types: regression and classification. Regression deals with the process involving continuous output, whereas classification deals with the process involving discrete (categorical) output.

Under supervised learning, one is given a data set $\mathcal{D}$ containing n observations of input-output $(\boldsymbol{x}_i, y_i)$, where $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, $\forall\ i \in \{0,1,\ldots,N-1\}$. This is a training data set involving a process called regression as one deals with continuous output. Given this training data set, the machine is trained to predict new inputs which are not listed in $\mathcal{D}$. The goal of supervised learning is to induce a function from observations on the training data set.

Gaussian processes form powerful models for regression problems. They have found a wide range of applications: robotics, data mining, geophysics, climate modeling, etc. (see [4], and references therein). Any Gaussian distribution is fully characterized by its mean and covariance function. The problem of learning in a Gaussian process is precisely the problem of finding suitable properties of the covariance function. In general, when only classical systems and strategies are in use, the implementation of a Gaussian process regression model with n training points typically requires $O(n^3)$ basic operations [4].

The application of principles in quantum mechanics has led to the realization of technologies in information processing and computation that can never be achieved within the realm of classical mechanics, such as teleportation and quantum key distribution [5–7]. Significantly, quantum computers are expected to have advantages over classical computers [5]. In theoretical computer science, quantum algorithms have been developed showing a significant advantage over their classical counterparts in terms of, in the best case scenario, an exponential speedup [8]. Importantly, quantum algorithms can also have a significant impact on machine learning, and this has led to the emergence of quantum machine learning [9].

The Harrow-Hassidim-Lloyd (HHL) algorithm, introduced in [10], gives a quantum algorithm for solving systems of linear equations. Specifically, let the system of linear equations be $A\boldsymbol{x} = \boldsymbol{b}$, where A is a matrix, and $\boldsymbol{b}$ a vector, and the goal is to find the vector $\boldsymbol{x}$. In [10], the case was considered in which one needs to know the expectation value of some operator associated with $\boldsymbol{x}$, e.g., $\boldsymbol{x}^\dagger M\boldsymbol{x}$ for a given matrix M, instead of the solution $\boldsymbol{x}$ itself. Assuming A is a sparse $N \times N$ matrix with condition number (ratio of largest and smallest eigenvalue) κ, classical algorithms can find $\boldsymbol{x}$ and estimate $\boldsymbol{x}^\dagger M\boldsymbol{x}$ in $O(N\sqrt{\kappa})$ time. However, in [10] the authors presented a quantum algorithm that ran in poly$(\log N, \kappa, 1/\epsilon)$ time, with ϵ precision in the output state (N.B., if one, in certain cases, avoids phase estimation the precision can be poly$[\log(1/\epsilon)]$ [11]). They showed that when the sparseness parameter of the matrix does not scale faster than polylogarithmically in

N, an exponential speedup is possible with the quantum linear systems algorithm. Recently, this quantum algorithm was applied to Gaussian process regression [12]. One of the contexts in which sparsely constructed Gaussian processes find applications is the problem involving inference in large data sets [13]. However, all applications so far have been limited to qubit or discrete-variable (DV) quantum systems.

Continuous-variable (CV) quantum systems are characterized by having an infinite-dimensional Hilbert space, and measurements involving observables with continuous eigenspectra [14]. A CV generalization of any DV quantum-system-assisted algorithm is essential in the context of developing algorithms for quantum computers involving CV systems, e.g., optical quantum computing [14]. The usefulness of CV quantum machine learning [15] goes beyond the processing of classical data sets that involve a discrete number of data. The output by the universal CV quantum computation is a CV state that evolves under a designed Hamiltonian [16]. The DV machine-learning subroutines are inefficient (incapable) at processing full CV states by themselves. This deficit of DV quantum-systems-assisted machine-learning subroutines can be curbed using predominantly CV quantum systems along with qubits when needed [16].

In this paper, we apply the techniques developed in [15] to generalize the DV quantum-assisted Gaussian process regression [12] to CV systems. For our task, we also describe an encoding method of a covariance matrix that gives a technique for a CV quantum-assisted singular value decomposition method of nonsparse low-rank matrices [17]. Furthermore, we consider the practical case of finite squeezing analysis for our algorithm.

Our discussion is organized as follows. We first introduce our notation and basic definitions in Sec. II. In Sec. III, we introduce a CV method of quantum singular value decomposition of nonsparse low-rank matrices. In Sec. IV, we illustrate a CV quantum-system-assisted algorithm for a Gaussian process regression model by efficiently computing its mean in Sec. IV A and covariance function in Sec. IV B. We base the algorithm on a scheme to encode the covariance matrix in an oracular setting for an efficient computation of the mean and covariance functions using CV quantum systems. Finally, in Sec. V, we give concluding remarks.

II. CLASSICAL GAUSSIAN PROCESS REGRESSION

In this review section, we introduce our notation and basic definitions that are needed for the discussion of Gaussian process regression. Let $\mathcal{N}(\boldsymbol{x}|\boldsymbol{m},\sigma^2)$ denote the Gaussian (normal) distribution of the variable $\boldsymbol{x}$ with mean $\boldsymbol{m} = \mathbb{E}[\boldsymbol{x}]$ and variance $\mathbb{V}[\boldsymbol{x}] = \mathbb{E}[(\boldsymbol{x}-\boldsymbol{m})^2] = \sigma^2$. Consider a training set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=0}^{N-1}$ of N d-dimensional inputs (input vectors) $\boldsymbol{x}_i$ and scalar outputs (or target values) y_i ($i \in \{0,1,\ldots,N-1\}$). The outputs y_i are accumulated together to form entries of an output N-dimensional vector $\boldsymbol{y}$. Furthermore, we assume that the outputs are noisy, i.e.,

$$y_i = f(\boldsymbol{x}_i) + \varepsilon, \tag{1}$$

where $f(\boldsymbol{x}_i)$ is the latent function [4] and $\varepsilon \sim \mathcal{N}(0,\sigma^2)$ denotes independent and identically distributed Gaussian noise.

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. A Gaussian process is completely specified by its mean and covariance function (kernel) [4], which for a real process $f(\boldsymbol{x})$ is defined by

$$k(\boldsymbol{x},\boldsymbol{x}') = \mathbb{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))], \tag{2}$$

where $m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})]$ denotes the mean function of $f(\boldsymbol{x})$. Let us denote the real process as

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}),k(\boldsymbol{x},\boldsymbol{x}')). \tag{3}$$

Given a new input (test point) $\boldsymbol{x}_*$, our goal is to predict the distribution of

$$f_* = f(\boldsymbol{x}_*). \tag{4}$$

One can consider an array of test points; however, for simplicity we consider only a single test point. The procedure described for a single test point can be simply generalized to multiple test point instances.

To this end, we note that the joint distribution of the observed target values and the function value at the test location are, respectively,

$$\begin{bmatrix} \boldsymbol{y} \\ f_* \end{bmatrix} \sim \mathcal{N}(\boldsymbol{0},\mathbb{K}), \qquad \mathbb{K} = \begin{bmatrix} K & \boldsymbol{k}_* \\ \boldsymbol{k}_*^T & k_{**} \end{bmatrix}, \tag{5}$$

where K is the $N \times N$ matrix with entries

$$K_{ij} = k(\boldsymbol{x}_i,\boldsymbol{x}_j) + \sigma^2\delta_{ij}, \tag{6}$$

and the entries of the vector $\boldsymbol{k}_* \equiv [\boldsymbol{k}_{*i}]$ are the covariance functions $k(\boldsymbol{x}_i,\boldsymbol{x}_*)$, and $k_{**} = k(\boldsymbol{x}_*,\boldsymbol{x}_*)$. Without loss of generality, we set the mean of the distribution to zero.

Using

$$\mathbb{K}^{-1} = \begin{bmatrix} K^{-1} + \tilde{k}_{**}^{-1}\tilde{\boldsymbol{k}}_*\tilde{\boldsymbol{k}}_*^T & -\tilde{k}_{**}^{-1}\tilde{\boldsymbol{k}}_* \\ -\tilde{k}_{**}^{-1}\tilde{\boldsymbol{k}}_*^T & \tilde{k}_{**}^{-1} \end{bmatrix}, \tag{7}$$

where

$$\tilde{\boldsymbol{k}}_* = K^{-1}\boldsymbol{k}_*, \qquad \tilde{k}_{**} = k_{**} - \boldsymbol{k}_* \cdot \tilde{\boldsymbol{k}}_*, \tag{8}$$

we deduce the conditional probability to be

$$P(f_*|\boldsymbol{y}) \sim \mathcal{N}(\boldsymbol{y} \cdot \tilde{\boldsymbol{k}}_*, \tilde{k}_{**}). \tag{9}$$

The task at hand boils down to the efficient computation of the mean $\boldsymbol{y} \cdot \tilde{\boldsymbol{k}}_*$ and variance $\tilde{k}_{**}$.

III. QUANTUM SINGULAR VALUE DECOMPOSITION METHOD OF NONSPARSE LOW-RANK MATRICES

In this section, we introduce a CV version of the quantum-assisted singular value decomposition method of nonsparse low-rank matrices which was first introduced for qubits in Ref. [17]. These results will form a subroutine in the next section for the quantum Gaussian process regression algorithm.

We begin by assuming for simplicity that $N = 2^n$ and the matrix K can be encoded as

$$\hat{K} = \begin{bmatrix} K & \\ & \mathbb{I} \end{bmatrix} \tag{10}$$

in an ensemble of $n+1$ qubits and accessed via oracle calls. For the oracle calls, we make use of the method for nonsparse

matrices in an oracular setting which requires only one-sparse simulation techniques [17].

To record and access the matrix $\hat{K}$, we first create the one-sparse Hermitian matrix [17]

$$H = \sum_{x,y=0}^{2N-1} \langle x|\hat{K}|y\rangle |x\rangle\langle y| \otimes |y\rangle\langle x| \tag{11}$$

whose entries are real numbers. This enlarges the Hilbert space quadratically, but because the matrix H has a single nonvanishing element in each row, its dynamics can be efficiently approximated. Indeed, we approximate each nonvanishing element of H by $2\zeta[\frac{\langle x|\hat{K}|y\rangle}{2\zeta}]$, where $[l]$ denotes the integer part of l. The resulting approximate matrix

$$\widetilde{H} = 2\sum_{x,y=0}^{2N-1} \left[\frac{\langle x|\hat{K}|y\rangle}{2\zeta}\right] |x\rangle\langle y| \otimes |y\rangle\langle x| \tag{12}$$

has entries which are even integers, and $H \approx \zeta\widetilde{H}$, i.e., $\|H - \zeta\widetilde{H}\| \lesssim \zeta$ [18]. It can be easily decomposed into a sum of matrices each of which has eigenvalues ± 1,

$$\widetilde{H} = \sum_{j=1}^{j_{\max}} H_j, \tag{13}$$

encoding efficiently a good approximation to all the information in the matrix $\hat{K}$. The sum in (13) contains at most $O(\|\widetilde{H}\|) \sim O(\lambda_{\max}/\zeta)$ terms, where $\lambda_{\max}$ is the largest eigenvalue of K. If $\lambda_{\max}$ is independent of the number of qubits, then the complexity of our quantum calculation is independent of N. This is not always the case [19], and depends on data distribution. We will discuss the complexity of the algorithm for a given error and the restrictions on the matrix K imposed by requiring exponential speedup further in Sec. IV.

We form the oracle calls to access H,

$$Q = i\sum_{j=1}^{j_{\max}} |j\rangle\langle j| \otimes e^{-\iota(\pi/2)H_j}. \tag{14}$$

Notice that Q is Hermitian as well as unitary, therefore $Q^2 = \mathbb{I}$.

Next we consider how to prepare states containing encodings of $|\boldsymbol{y}\rangle$ and $|\boldsymbol{k}_*\rangle$. We normally encode a N-dimensional unit vector $\boldsymbol{v}$ by forming the n-qubit state $|\boldsymbol{v}\rangle = \sum_i v_i|i\rangle$. However, the vectors we are interested in are not unit vectors, and we are also interested in the signs of inner products, not just their absolute values. To encode this additional information, e.g., for $\boldsymbol{y}$, we shall use $n+1$ qubits and form the unit vector corresponding to the $2N$-dimensional vector $[y_i/c(y), \sqrt{1-y_i^2/c^2(y)}]$, where $c(y) > y_i$, for all $i \in \{0,1,\ldots,N-1\}$, and encode it on the $(n+1)$-qubit state

$$|\boldsymbol{y}\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}\left(\frac{y_i}{c(y)}|i\rangle + \sqrt{1-\frac{y_i^2}{c^2(y)}}|N+i\rangle\right). \tag{15}$$

One way to prepare such a state is to form a unitary (sequence of rotations) U_y such that $U_y|0\cdots 0\rangle = |\boldsymbol{y}\rangle$. This can be done efficiently, as long as the components of $\boldsymbol{y}$ are relatively uniform [19]. In a similar way, we can encode $\boldsymbol{k}_*$ in the $(n+1)$-qubit state

$$|\boldsymbol{k}_*\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}\left(\frac{k_{*i}}{c(k_*)}|i\rangle + \sqrt{1-\frac{k_{*i}^2}{c^2(k_*)}}|N+i\rangle\right). \tag{16}$$

IV. QUANTUM GAUSSIAN PROCESS REGRESSION

In this section, we illustrate our quantum algorithm using quantum CV systems to implement the task of efficiently computing the mean $\boldsymbol{y}\cdot\tilde{\boldsymbol{k}}_*$ and variance $\tilde{k}_{**}$ for a Gaussian process regression model. We do so by including our previous results of the quantum singular value decomposition as a subroutine.

A. Efficient computation of mean

Given the N-dimensional vectors $\boldsymbol{y}$ and $\boldsymbol{k}_*$ defined above, we form the $4N$-dimensional vector $[y_i/c(y), \sqrt{1-y_i^2/c^2(y)}, k_{*i}/c(k_*), \sqrt{1-k_{*i}^2/c^2(k_*)}]$, and encode it in the $(n+2)$-qubit state $|\boldsymbol{y},\boldsymbol{k}_*\rangle$, as outlined above. This can be done efficiently with a string of unitary operations, provided the components are relatively uniform [19], or they have been encoded and stored in a qRAM [20,21] by a third party.

Evidently, our input state can be written in terms of the states (15) and (16) defined above as

$$|\boldsymbol{y},\boldsymbol{k}_*\rangle \equiv \frac{1}{\sqrt{2}}(|\boldsymbol{y}\rangle|0\rangle + |\boldsymbol{k}_*\rangle|1\rangle). \tag{17}$$

To this state we append two CV resource modes in the squeezed state

$$|\Phi_R(\xi)\rangle = \frac{1}{\sqrt{\pi}\xi}\int dq_R d\tilde{q}_R e^{-(1/2\xi^2)[q_R^2+\tilde{q}_R^2]}|q_R\rangle|\tilde{q}_R\rangle, \tag{18}$$

written in terms of the q quadratures, q_R and $\tilde{q}_R$, respectively, of the resource modes. It is advantageous to make the squeezing parameter ξ as small as technologically feasible, thus forming the state

$$|\chi[\boldsymbol{y},\boldsymbol{k}_*,\Phi_R(\xi)]\rangle \equiv |\boldsymbol{y},\boldsymbol{k}_*\rangle|\Phi_R(\xi)\rangle. \tag{19}$$

Next we apply the unitary

$$\mathcal{U} = e^{\iota\gamma(\hat{K}/4N)\hat{N}p_R\tilde{p}_R}, \tag{20}$$

where $\hat{N} = \frac{\mathbb{I}-Z}{2}$ is a projection acting on the last qubit of our state (17), p_R and $\tilde{p}_R$ are p quadrature operators acting on the resource modes, and γ is a parameter that can be adjusted at will. To implement it, instead of applying an evolution involving $\hat{K}$, we use the quadratically enlarged matrix H [Eq. (12)] which contains the same information, but leads to simpler dynamics. We will then use the resulting unitary as a generalized exponential swap to apply the desired matrix $\hat{K}$.

To implement the approximation to H and $\widetilde{H}$ [Eqs. (12) and (13)], we need a "fractional" query, $Q^{\delta p_R\tilde{p}_R\hat{N}}$, where δ is an arbitrary real number. Notice that

$$Q^{\delta p_R\tilde{p}_R\hat{N}} = \tfrac{1}{2}(\mathbb{I}+Q)\otimes e^{\iota(\pi\delta/2)p_R\tilde{p}_R\hat{N}} + \tfrac{1}{2}(\mathbb{I}-Q)\otimes e^{-\iota(\pi\delta/2)p_R\tilde{p}_R\hat{N}}, \tag{21}$$

acting on, say, $|\psi\rangle|\phi\rangle$. We append an ancilla qubit in the state $|+\rangle_A$, and then perform a control Q on $|\psi\rangle$ with the ancilla as

control. We obtain

$$\frac{1}{\sqrt{2}}(|\psi\rangle|\phi\rangle|0\rangle_A + Q|\psi\rangle|\phi\rangle|1\rangle_A). \tag{22}$$

Then we rotate the ancilla so that $|0\rangle_A \to |+\rangle_A$, $|1\rangle_A \to |-\rangle_A$, which yields

$$\tfrac{1}{2}[(\mathbb{I}+Q)|\psi\rangle|\phi\rangle|0\rangle_A + (\mathbb{I}-Q)|\psi\rangle|\phi\rangle|1\rangle_A]. \tag{23}$$

Next we apply the unitary $e^{\iota(\pi\delta/2)p_R\tilde{p}_R\hat{N}Z_A}$, where Z_A is the Pauli matrix Z acting on the ancilla. It can be implemented with a non-Gaussian gate [15,22], if $|0\rangle$,$|1\rangle$ represent logical qubits realized by a pair of qumodes $|01\rangle = \hat{b}^\dagger|00\rangle$,$|10\rangle = \hat{a}^\dagger|00\rangle$, so that $Z_A = \hat{b}^\dagger\hat{b} - \hat{a}^\dagger\hat{a}$. We obtain

$$\frac{\mathbb{I}+Q}{2}e^{\iota(\pi\delta/2)p_R\tilde{p}_R\hat{N}}|\psi\rangle|\phi\rangle|0\rangle_A + \frac{\mathbb{I}-Q}{2}e^{-\iota(\pi\delta/2)p_R\tilde{p}_R\hat{N}}|\psi\rangle|\phi\rangle|1\rangle_A. \tag{24}$$

Finally, we perform a projective measurement on the ancilla projecting it onto $|+\rangle_A$, resulting in the desired state $Q^{\delta p_R\tilde{p}_R\hat{N}}|\psi\rangle|\phi\rangle$. This projection is successful 50% of the time, as is easily verified.

We can now implement

$$e^{-\iota\gamma H p_R\tilde{p}_R\hat{N}} \approx e^{-\iota\gamma\epsilon\tilde{H}p_R\tilde{p}_R\hat{N}} \approx \left(\prod_j e^{-\iota(\gamma\epsilon/M)H_j p_R\tilde{p}_R\hat{N}}\right)^M \tag{25}$$

using $Q^{\delta p_R\tilde{p}_R\hat{N}}$, where $\delta = \frac{2\gamma\epsilon}{\pi M}$. Let $|\psi\rangle = |1\rangle|\chi\rangle$, initially. Let P be any permutation matrix, so that by repeatedly acting with P on $|1\rangle$, we span all states $|j\rangle$, $j = 1,\ldots,j_{\max}$. Then

$$(Q^{\delta p_R\tilde{p}_R\hat{N}}\ P\otimes\mathbb{I})^{j_{\max}}|\psi\rangle = \mathbb{I}\otimes\prod_j e^{-\iota(\gamma/\epsilon M)H_j p_R\tilde{p}_R\hat{N}}|\psi\rangle. \tag{26}$$

Having constructed the unitary (26), we may implement the unitary (25) by repeating the above process M times. We will use this construction to implement the unitary (20) following [17]. Let $\rho = |\chi\rangle\langle\chi|$ [Eq. (19)] be the state on which (20) will act. We introduce the symmetric state

$$|s\rangle = \frac{1}{2\sqrt{N}}\sum_{x=0}^{2N-1}|x\rangle. \tag{27}$$

We then act on the state $|s\rangle\langle s|\otimes\rho$ with the unitary (26), and trace over the degrees of freedom of the register in which $|s\rangle$ resides. We obtain

$$\mathrm{tr}[e^{-\iota(\gamma/M)Hp_R\tilde{p}_R\hat{N}}\ |s\rangle\langle s|\otimes\rho\ e^{\iota(\gamma/M)Hp_R\tilde{p}_R\hat{N}}] \approx \mathcal{U}_M\rho\,\mathcal{U}_M^\dagger, \tag{28}$$

where $\mathcal{U}_M = e^{-\iota(\gamma/4MN)\hat{K}\hat{N}p_R\tilde{p}_R}$. Thus, the above procedure yielded an evolution involving the desired matrix $\hat{K}$ from the quadratically enlarged, but dynamically simpler, matrix H containing all the entries of $\hat{K}$ [Eq. (12)].

The error in (28) is $\epsilon_M \lesssim \frac{\gamma^2}{M^2\xi^4}\|\hat{K}\|^2$, where $\|\hat{K}\|$ is the magnitude of the largest matrix element of $\hat{K}$, and we used $|\lambda| \leqslant 4N\|\hat{K}\|$, where λ is any eigenvalue of $\hat{K}$.

By repeating this process M times, we arrive at the desired result (20) [since $\mathcal{U} \approx (\mathcal{U}_M)^M$]. The cumulative error is $\epsilon = M\epsilon_M \lesssim \frac{\gamma^2}{M\xi^4}\|\hat{K}\|^2$. For a large number of steps, only large enough eigenvalues contribute, specifically, $\gamma|\lambda|/N \gtrsim \xi^2$. Let us choose a small enough squeezing parameter ξ (restricted by current technology) and a large enough adjustable parameter γ, so that the error introduced by restricting to relatively large eigenvalues is ϵ, i.e.,

$$\gamma \sim \frac{\xi^2}{\epsilon}. \tag{29}$$

It should be noted that the smallest eigenvalue of $\hat{K}$ can also be controlled to a certain extent by increasing the variance noise $\sigma^2\mathbb{I}$, so that $\|\hat{K}\| \gtrsim \sigma^2$, which may relax the constraint (29) on γ.

It follows that the number of oracle calls required for the algorithm is $M \lesssim \frac{\gamma^2}{\epsilon\xi^4}\|\hat{K}\|^2 \sim \|\hat{K}\|^2/\epsilon^3$. If K is a low-rank matrix which is dense with relatively small matrix elements, then $\|\hat{K}\| \sim O(\mathrm{poly}\log N)$ [17]. Moreover, if we are interested in errors $1/\epsilon \sim O(\mathrm{poly}\log N)$, then also the number of oracle calls $M \sim O(\mathrm{poly}\log N)$.

The above considerations constrain the parameters ξ and γ to be in a range that facilitates an accurate calculation of the mean. Indeed, we obtain

$$\mathcal{U}|\chi\rangle = \frac{1}{\sqrt{2}}(|\boldsymbol{y}\rangle|0\rangle + e^{\iota\gamma(\hat{K}/4N)p_R\tilde{p}_R}|\boldsymbol{k}_*\rangle|1\rangle)|\Phi_R(\xi)\rangle. \tag{30}$$

Next, we measure the q quadrature, q_R and $\tilde{q}_R$, respectively, of the resource modes. If the outcome is $(q_R,\tilde{q}_R)$ with $|q_R|,|\tilde{q}_R| \lesssim \xi$, then the state is projected onto $\Pi_\xi\mathcal{U}|\chi\rangle$, where

$$\Pi_\xi = \int_{-\xi}^{\xi}dq_R\int_{-\xi}^{\xi}d\tilde{q}_R|q_R,\tilde{q}_R\rangle\langle q_R,\tilde{q}_R|. \tag{31}$$

As shown below, this results in a state which is independent of the resource measurement outcomes to a good approximation. Therefore, the probability that the resource measurement successfully implements Π_ξ is

$$\left(\frac{1}{\sqrt{\pi}\xi}\int_{-\xi}^{\xi}dqe^{-q^2/\xi^2}\right)^2 = \mathrm{erf}^2(1), \tag{32}$$

or numerically 71%.

After a straightforward calculation, we obtain

$$\langle q_R,\tilde{q}_R|e^{\iota\gamma(\hat{K}/4N)\hat{N}p_R\tilde{p}_R}|\Phi_R(\xi)\rangle \propto \frac{e^{-[\xi^2(q_R^2+\tilde{q}_R^2)+2\iota\gamma q_R\tilde{q}_R(\hat{K}/4N)\hat{N}/2(\xi^4+\gamma^2(\hat{K}^2/16N^2)\hat{N}^2)]}}{\sqrt{\xi^4+\gamma^2\frac{\hat{K}^2}{16N^2}\hat{N}^2}}. \tag{33}$$

where the remaining operators $\hat{K}$ and $\hat{N}$ act on the Hilbert space of the state of our system (17), with $\hat{N}$ acting on its last qubit and $\hat{K}$ on the rest of the qubits. Choosing ξ to be small enough and γ to be large enough, as outlined above, we may approximate

$$\frac{\langle q_R,\tilde{q}_R|e^{\iota\gamma\hat{K}p_R\tilde{p}_R}|\Phi_R(\xi)\rangle}{\langle q_R,\tilde{q}_R|\Phi_R(\xi)\rangle} \approx \frac{\xi^2}{\gamma}\left(\frac{\hat{K}}{4N}\right)^{-1}. \tag{34}$$

The resource modes decouple, and the remaining projected state $|\hat{\chi}(\boldsymbol{y},\boldsymbol{k}_*,s)\rangle$ is approximately

$$|\hat{\chi}\rangle \approx |\boldsymbol{y}\rangle|0\rangle + \frac{\xi^2}{\gamma}\hat{K}^{-1}|\boldsymbol{k}_*\rangle|1\rangle. \tag{35}$$

Next, we measure Z on the first qubit of the $n+1$ qubit system and X on the appended (last) qubit. The expectation value of $\frac{\mathbb{I}+Z}{2}\otimes X$ for the state $|\hat{\chi}\rangle$ is

$$\langle\hat{\chi}|\frac{\mathbb{I}+Z}{2}\otimes X|\hat{\chi}t\rangle = \frac{2\xi^2}{Nc(y)c(k_*)\gamma}\boldsymbol{y}^T K^{-1}\boldsymbol{k}_* = \frac{2\xi^2\boldsymbol{y}\cdot\tilde{\boldsymbol{k}}_*}{Nc(y)c(k_*)\gamma}, \tag{36}$$

from which we easily deduce the mean $\boldsymbol{y}\cdot\tilde{\boldsymbol{k}}_*$:

$$\boldsymbol{y}\cdot\tilde{\boldsymbol{k}}_* = \frac{N\gamma}{2\xi^2}c(y)c(k_*)\langle\hat{\chi}|\frac{\mathbb{I}+Z}{2}\otimes X|\hat{\chi}\rangle. \tag{37}$$

B. Efficient computation of variance

To calculate the variance, we need to be able to efficiently compute $\boldsymbol{k}_*\cdot\tilde{\boldsymbol{k}}_* = \boldsymbol{k}^T K^{-1}\boldsymbol{k}_*$ (8) as k_{**} is given. The calculation of $\boldsymbol{k}_*\cdot\tilde{\boldsymbol{k}}_*$ follows the same lines of the calculation of the mean with $\boldsymbol{y}$ replaced by $\boldsymbol{k}_*$ in (17) and (19). Because of this replacement, we need to consider the $(n+1)$-qubit state $|\boldsymbol{k}_*\rangle$ corresponding to a $2N$-dimensional vector.

To this system, we append a qubit in the state $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$ as well as two resource modes in the squeezed state (18), thus forming the state

$$|\chi(\boldsymbol{k}_*,\boldsymbol{k}_*,\Phi(\xi))\rangle = \frac{1}{\sqrt{2}}|\boldsymbol{k}_*\rangle(|0\rangle+|1\rangle)|\Phi_R(\xi)\rangle. \tag{38}$$

Following the same steps as that for the calculation of the mean, we obtain

$$\langle\hat{\chi}|\frac{\mathbb{I}+Z}{2}\otimes X|\hat{\chi}\rangle = \frac{2\xi^2}{N\gamma c^2(k_*)}\boldsymbol{k}_*^T K^{-1}\boldsymbol{k}_* = \frac{2\xi^2\boldsymbol{k}_*\cdot\tilde{\boldsymbol{k}}_*}{N\gamma c^2(k_*)}, \tag{39}$$

from which we easily deduce the variance $\tilde{k}_{**}$:

$$\tilde{k}_{**} = k_{**} - \frac{N\gamma}{2\xi^2}c^2(k_*)\langle\hat{\chi}|\frac{\mathbb{I}+Z}{2}\otimes X|\hat{\chi}\rangle. \tag{40}$$

Given that both $\boldsymbol{y}$ and $\boldsymbol{k}_*$ are sparse, and K is well conditioned, we have an efficient way of computing both the mean and variance function of Gaussian process regression. In this situation, because we are making use of a quantum linear systems algorithm, we achieve an exponential speedup over its classical counterpart.

V. CONCLUSION

We presented a continuous-variable quantum-system-assisted Gaussian process regression algorithm that offers the potential of an exponential speedup over classical techniques. It generalized the result given in [12] where the authors had initially considered the application of quantum systems of linear equations algorithm [10] to Gaussian process regression using discrete-variable quantum systems. The application of such HHL algorithm constrains the matrix K (6) related to Gaussian processes to be well conditioned. K needs to be robustly invertible [19], which restricts the condition number κ to remain low even as N increases. We can make K robust by increasing the variance noise ($\sigma^2\mathbb{I}$) so that $\lambda_{\min}$ remains above a certain threshold ($\lambda_{\min}\gtrsim\sigma^2$). This dilution trick would work only if the statistical properties of the concerned model are not significantly altered.

In Ref. [17], the authors provided a method for nonsparse matrices in an oracular setting which required only one-sparse simulation techniques. We made use of this method to encode K for the computation of the mean and covariance function of a Gaussian process regression model. Our presented method provides a continuous-variable quantum-assisted singular value decomposition of nonsparse low-rank matrices. This hints at applications of our technique to subroutines beyond quantum systems of linear equations algorithms.

ACKNOWLEDGMENTS

We thank Patrick Rebentrost for helpful comments and feedback. S.D. acknowledges support from the LSU Graduate School Economic Development Assistantship. G.S. acknowledges support from the U.S. Office of Naval Research under Award No. N00014-15-1-2646.

[1] A. L. Samuel, Some studies in machine learning using the game of checkers, IBM J. Res. Dev. **3**, 210 (1959).
[2] M. W. Libbrecht and W. S. Noble, Machine learning applications in genetics and genomics, Nat. Rev. Genet. **16**, 321 (2015).
[3] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, Towards fully autonomous driving: Systems and algorithms, in *Intelligent Vehicles Symposium (IV)* (IEEE, New York, 2011), pp. 163–168.
[4] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)* (MIT, Cambridge, MA, 2005).
[5] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, United Kingdom, 2000).
[6] C. H. Bennett and D. P. DiVincenzo, Quantum information and computation, Nature (London) **404**, 247 (2000).
[7] J. P. Dowling, *Schrödinger's Killer App: Race to Build the World's First Quantum Computer* (CRC Press, Boca Raton, FL, 2013).
[8] A. Montanaro, Quantum algorithms: An overview, npj Quantum Inf. **2**, 15023 (2016).
[9] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).
[10] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**, 150502 (2009).
[11] A. M. Childs, R. Kothari, and R. D. Somma, Quantum linear systems algorithm with exponentially improved dependence on precision, SIAM J. Comput. **46**, 1920 (2017).
[12] Z. Zhao, J. K. Fitzsimons, and J. F. Fitzsimons, Quantum assisted Gaussian process regression, arXiv:1512.03929.

[13] A. Melkumyan and F. Ramos, A sparse covariance function for exact Gaussian process inference in large datasets, in IJCAI 2009 (unpublished), Vol. 9, pp. 1936–1042.

[14] C. Weedbrook, S. Pirandola, R. García-Patrón, N. J. Cerf, T. C. Ralph, J. H. Shapiro, and S. Lloyd, Gaussian quantum information, Rev. Mod. Phys. **84**, 621 (2012).

[15] H.-K. Lau, R. Pooser, G. Siopsis, and C. Weedbrook, Quantum Machine Learning Over Infinite Dimensions, Phys. Rev. Lett. **118**, 080501 (2017).

[16] S. Lloyd and S. L. Braunstein, Quantum Computation Over Continuous Variables, Phys. Rev. Lett. **82**, 1784 (1999).

[17] P. Rebentrost, A. Steffens, and S. Lloyd, Quantum singular value decomposition of non-sparse low-rank matrices, Phys. Rev. A **97**, 012327 (2018).

[18] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Exponential improvement in precision for simulating sparse Hamiltonians, in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing* (ACM, New York, 2014), pp. 283–292.

[19] S. Aaronson, Read the fine print, Nat. Phys. **11**, 291 (2015).

[20] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum Random Access Memory, Phys. Rev. Lett. **100**, 160501 (2008).

[21] V. Giovannetti, S. Lloyd, and L. Maccone, Architectures for a quantum random access memory, Phys. Rev. A **78**, 052310 (2008).

[22] K. Marshall, R. Pooser, G. Siopsis, and C. Weedbrook, Repeat-until-success cubic phase gate for universal continuous-variable quantum computation, Phys. Rev. A **91**, 032321 (2015).