Quantum-assisted Gaussian process regression

Zhikuan Zhao, Jack K. Fitzsimons, and Joseph F. Fitzsimons

Phys. Rev. A **99**, 052331 — Published 22 May 2019

DOI: 10.1103/PhysRevA.99.052331

Quantum assisted Gaussian process regression

Zhikuan Zhao,[1, 2] Jack K. Fitzsimons,[3] and Joseph F. Fitzsimons[1, 2, 4, *]

[1] *Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372*
[2] *Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, Singapore 117543*
[3] *Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK*
[4] *Horizon Quantum Computing, 79 Ayer Rajah Crescent, Singapore 139955*

Gaussian processes (GP) are a widely used model for regression problems in supervised machine learning. Implementation of GP regression typically requires $O(n^3)$ logic gates. We show that the quantum linear systems algorithm [Harrow et al., Phys. Rev. Lett. 103, 150502 (2009)] can be applied to Gaussian process regression (GPR), leading to an exponential reduction in computation time in some instances. We show that even in some cases not ideally suited to the quantum linear systems algorithm, a polynomial increase in efficiency still occurs.

I. INTRODUCTION

The enterprise of designing quantum algorithms has come a long way since Feynman's original vision of utilizing the exponential Hilbert space in quantum mechanics to simulate quantum physics [1]. Among the most celebrated results are Shors factoring algorithm [2] and Grover's search algorithm [3]. More recently, machine learning has emerged as a field in which quantum algorithms can have a dramatic impact [4–9]. Of particular interest to the field of machine learning, in 2009 Harrow *et al* presented a quantum algorithm which produces a superposition state $|\mathbf{x}\rangle$ with ϵ error, such that the vector $|\mathbf{x}\rangle$ solves the linear system $A|\mathbf{x}\rangle = |\mathbf{b}\rangle$ [10]. For an $n \times n$ s-sparse matrix A with condition number (the ratio between its largest and smallest eigenvalues) κ, the runtime roughly grows as $\tilde{O}(\log(n)\kappa^2 s^2/\epsilon)$ (where $\tilde{O}$ suppresses slower growing contribution), while the classical counter-part, matrix inversion algorithms for sparse matrices runs at $O(n\kappa s\log(1/\epsilon))$ using the conjugate gradient method [11].

Gaussian processes (GP) are commonly used as powerful models for regression problems in the field of supervised machine learning, and have been widely applied across a broad spectrum of applications, ranging from robotics, data mining, geophysics (where they are referred to as kriging), climate modelling and predicting price behaviour of commodities in financial markets. Recently, an intriguing correspondence between GP and deep neural networks has been established in [12]. Although GP models are becoming increasingly popular in the machine learning community, they are known to be computationally expensive, hindering their widespread adoption. A practical implementation of Gaussian process regression (GPR) model with n training points typically requires $O(n^3)$ basic operations [13]. This has lead to significant effort aimed at reducing the computational cost of working with such models, with investigations into low rank approximations of GPs [14], variational approximations [15] and Bayesian model combination for distributed GPs [16].

In this work we demonstrate that the Quantum linear system algorithm (QLSA) described in [10] can be used to dramatically speed up computation in GPR. We start by reviewing both the basics of classical GPR and its conventional implementation, as well as the original QLSA. We then propose a procedure of applying QLSA to GPR modelling, and discuss the performance of such a procedure. We will address the practical aspects of applying our procedure to specific GPR problems in terms of the potential caveats of QLSA summarised in [17].

II. GAUSSIAN PROCESS REGRESSION

Supervised machine learning endeavours to learn the relationship between the input and output of a system based on a set of examples, referred to as the observations of a training set. Gaussian processes offer a number of desirable properties in doing this such as ease in expressing uncertainty, the ability to model a wide range of behaviours under a simple parametrisation, and admitting a natural Bayesian interpretation.

Given a training set $\mathcal{T} = \{\mathbf{x}_i, y_i\}_{i=0}^{n-1}$, containing n d-dimensional inputs, $\{\mathbf{x}_i\}_{i=0}^{n-1}$, and corresponding outputs, $\{y_i\}_{i=0}^{n-1}$, we wish to model a latent function $f(\mathbf{x})$ such that

$$y = f(\mathbf{x}) + \epsilon_{\text{noise}}, \tag{1}$$

where $\epsilon_{noise} \sim \mathcal{N}(0, \sigma_n^2)$ is independent and identically distributed Gaussian noise. As such, given a new input (we will call this a "test point"), $\mathbf{x}_*$, we aim to have a predictive distribution for $f_* = f(\mathbf{x}_*)$.

The GP approach to such a regression problem models the behaviour of latent variables $\{f(\mathbf{x_i})\}_{i=0}^{n-1}$ as a joint multi-dimensional Gaussian distribution [13]. A GPR model is fully specified by a mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and a covariance function (also known as kernel) $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$, where $\mathbb{E}[z]$ denotes the expectation value of z. Without loss of generality we can assume the function being modelled to have zero prior mean. Conditioning on the training data set, we write the predictive distribution of f_* in the form of a multi-variable Gaussian distribution [18],

$$p(f_*|\mathbf{x}_*, \mathcal{T}) \sim \mathcal{N}(\bar{f}_*, \mathbb{V}[f_*]). \tag{2}$$

The central goal in GPR models is to predict the mean of this distribution, also known as the mean predictor $\bar{f}_*$, and its variance $\mathbb{V}[f_*]$ given the inputs $\{\mathbf{x}_i\}$, observed output vector $\mathbf{y}$,

[*] Electronic address: joe.fitzsimons@nus.edu.sg

the covariance functions k, and the noise variance σ_n^2. For simplicity, we consider only one test point, although the same principle applies to an array of test points. Let the entries of the vector $\mathbf{k}_*$ denotes the covariance functions between the test point $\mathbf{x}_*$ and each of the n input points in the training set, such that $\mathbf{k}_* = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}_*) - m(\mathbf{x}_*))]$. We denote by K the $n \times n$ matrix of covariance functions between the input points in the training set. Following the derivation presented in [13], we obtain the moments of a zero mean GP as

$$\bar{f}_* = \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{y} \tag{3}$$

$$\mathbb{V}[f_*] = k\left(\mathbf{x}_*, \mathbf{x}_*\right) - \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_*. \tag{4}$$

We will outline a typical implementation of GPR on a classical machine: The first step is to compute the Cholesky decomposition of $(K + \sigma_n^2 I)$, that is, to find the lower-triangular matrix L, known as the Cholesky factor, such that $(K + \sigma_n^2 I) = LL^T$. The computation of the Cholesky factor is known to be numerically stable, and has runtime proportional to n^3. Writing $\boldsymbol{\alpha} = (K + \sigma_n^2 I)^{-1}\mathbf{y}$, the mean predictor is rewritten as $\bar{f}_* = \mathbf{k}_*^T \boldsymbol{\alpha}$. Computing α then amounts to solving $LL^T\boldsymbol{\alpha} = \mathbf{y}$. Using the backslash notation $L\backslash\mathbf{y}$ to denote the vector $\mathbf{y}'$ which solves the triangular linear system $L\mathbf{y}' = \mathbf{y}$, $\boldsymbol{\alpha} = L^T\backslash L\backslash\mathbf{y}$ can be computed by solving two triangular systems, taking time proportional to n^2. Similarly, $\mathbb{V}[f_*] = k\left(\mathbf{x}_*, \mathbf{x}_*\right) - (L\backslash\mathbf{k}_*)^T(L\backslash\mathbf{k}_*)$ can be computed with a number of basic arithmetic operations proportional to n^2. Note that the $k\left(\mathbf{x}_*, \mathbf{x}_*\right)$ term is simply the covariance of the test point with itself, and only costs a constant time to compute. Therefore the total runtime of computing the mean and variance for GPR amounts to $O(n^3)$. For problems involving thousands of input points, the exact inference in GPR become intractable, which motivates the search for a quantum approach to accelerate this computation. This is where the QLSA, first introduced in [10], offers an advantage.

III. QUANTUM LINEAR SYSTEM ALGORITHM

We now give an outline of the original QLSA to solve the linear system, $A\mathbf{x} = \mathbf{b}$:

- Prepare the state $|\mathbf{b}\rangle = (\mathbf{b}^T\mathbf{b})^{-1/2} \sum\limits_{i=0}^{n-1} b_i |i\rangle$ to encode the vector $\mathbf{b}$. Prepare an ancilla register in a superposition state $\frac{1}{\sqrt{T}} \sum_{\tau=0}^{T} |\tau\rangle$.

- Simulate A as a Hamiltonian at time τ applied to $|\mathbf{b}\rangle$ using phase estimation techniques described in [19], and expand $|\mathbf{b}\rangle$ into the eigenbasis of A. After this evolution, we obtain the state

$$|\phi_1\rangle = \frac{1}{\sqrt{T}} \sum_{i=0}^{n-1} \sum_{\tau=0}^{T-1} |\tau\rangle e^{i\lambda_i t_0 \tau / T} \beta_i |\mu_i\rangle, \tag{5}$$

where λ_i are the eigenvalues and $|\mu_i\rangle$ are the eigenvectors of A . Each $|\mu_i\rangle$ is associated with a complex amplitude β_i. The time t_0 here scales linearly with the condition number κ. The time period in the second summation is chosen to be some large T as in the improved phase-estimation procedure described in [20].

- Apply the quantum Fourier transform (QFT) [21] to the first register in $|\phi_1\rangle$ and obtain

$$|\phi_2\rangle = \sum_{i=0}^{n-1} \beta_i |\lambda_i\rangle |\mu_i\rangle. \tag{6}$$

- Introduce an ancilla qubit and perform a controlled rotation on it to yield the extended state

$$|\phi_3\rangle = \sum_{i=0}^{n-1} \beta_i |\lambda_i\rangle |\mu_i\rangle \left(\sqrt{1 - \frac{c^2}{\lambda_i^2}} |0\rangle + \frac{c}{\lambda_i} |1\rangle \right). \tag{7}$$

Here the constant c is used to ensure that the rotation angle is bounded by π.

- Finally, reverse the phase estimation to uncompute $|\lambda_i\rangle$. Measure the ancilla qubit. A result of $|1\rangle$ results in the state vector encoding the solution of $A\mathbf{x} = \mathbf{b}$,

$$|\mathbf{x}\rangle = |\phi_{final}\rangle = \sum_{i=0}^{n-1} \frac{\beta_i}{\lambda_i} |\mu_i\rangle \tag{8}$$

For simplicity, we have omitted global normalisation factors in the last step of the above outline, and have assumed A to be Hermitian throughout. However, Ref. [10] also includes a treatment to "Hermitianize" a general A, which involves building an anti-diagonal block matrix with the elements of $A^\dagger$ and A in the lower and upper half of the new matrix respectively. Once $|\mathbf{x}\rangle$ has been produced, quantum measurements can be used to estimate expectation values corresponding to some desired quantity of the form $\langle x|M|x\rangle$. For a quick account of the runtime, we note that the ϵ-error runtime in phase estimation scales quadratically with the sparseness s of A [19], $t_0 = O(\kappa/\epsilon)$, the repetition needed to obtain the desired measurement on the ancilla qubit scales proportionally to κ, and hence the total runtime amounts to $\tilde{O}(\log(n)\kappa^2 s^2/\epsilon)$. We direct interested readers to Ref. [10] and its supplementary material for a detailed error and runtime analysis.

Soon after the original QLSA was proposed, Clader *et al* extended the algorithm to include an efficient method to prepare the input encoding using entangled states in $O(1)$ query complexity with the help of an oracle which calculates the amplitude and phase components of the vector $|\mathbf{b}\rangle$ [22]. In the same paper, the authors also developed a scheme based on the Sparse Approximate Inverse Preconditioners (SAIP) technique [23, 24] to precondition A which incurs an overhead of $O(s^3)$. This results in a modified linear system in which the matrix is well conditioned. Childs *et al* further modified the QLSA based on implementing operators with Fourier or Chebyshev series representations, which further suppressed the runtime through a logarithmic ϵ-precision dependence [25]. The sparse-dependent efficiency of the Hamiltonian simulation stage in the QLSA was also subsequently improved [26], leading to a new runtime which scales as $\tilde{O}(\log(n)\kappa^2 s/\epsilon)$. In the context of dense linear systems, a recent result of [27] shows an alternative linear system solver that circumvents the Hamiltonian simulation step, achieving a runtime scaling of $\tilde{O}(\sqrt{n}\log(n)\kappa^2/\epsilon)$

Despite the promising exponential speed-up QLSA can potentially provide, one has to apply it with care. As noted by Aaronson [17], there are four practical areas that need particular care in any application of the original QLSA: (1) The time taken to prepare $|\mathbf{b}\rangle$ encoding $\mathbf{b}$ needs to be taken into account; (2) the matrix A has to be robustly invertible, κ needs to grow at most polylogarithmically in n to maintain an exponential speed-up; (3) one also needs to address the sparseness contribution to the total runtime, since the general phase estimation sub-routine in QLSA costs polynomial time in s; (4) although the output of QLSA is the state $|\mathbf{x}\rangle$, there is no efficient way to extract entries of the vector $\mathbf{x}$. One needs to make sure that the matter of practical interest does not span the full glory of $\mathbf{x}$, but is restricted only to information which is accessible with relatively few copies of $|\mathbf{x}\rangle$. For example, one can efficiently estimate quantities such as $\langle\mathbf{x}|M|\mathbf{x}\rangle$, where M is some Hermitian matrix of interest which can be efficiently implemented as an observable, since this simply amounts to the expectation value of the observable M on $|\mathbf{x}\rangle$. We now introduce a procedure for applying QLSA to Gaussian process regression, and then address each of these practicality concerns.

IV. QUANTUM GAUSSIAN PROCESS ALGORITHM

We observe from equations 3 and 4 that the computation of f_* and $\mathbb{V}[f_*]$ involves solving linear systems of the forms $(K+\sigma_n^2 I)\boldsymbol{\alpha} = \mathbf{y}$ and $(K+\sigma_n^2 I)\boldsymbol{\eta} = \mathbf{k}_*$ respectively, where $\mathbf{k}_*^T\boldsymbol{\alpha} = \bar{f}_*$ and $k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T\boldsymbol{\eta} = \mathbb{V}[f_*]$. The common linear structure suggests that we can apply QLSA to extract useful information.

In order to compute these values, we will extend the quantum linear systems algorithm in two ways. First, we need an efficient method to prepare a state $|\mathbf{v}\rangle$ from a classical representation of a vector $\mathbf{v}$ of length n. To achieve this we use an approach based on quantum random access memory (QRAM) [28], which we modify to allow preparation of sparse (or approximately sparse) vectors. To prepare a state corresponding to the $s_{\mathbf{v}}$-sparse vector $\mathbf{v}$ with entries v_i, a register is prepared in a superposition $s_{\mathbf{v}}^{-1/2}\sum_{i:v_i\neq 0}|i\rangle\otimes|0\rangle$. The QRAM acts as an oracle that performs the mapping:

$$s_{\mathbf{v}}^{-1/2}\sum_{i:v_i\neq 0}|i\rangle\otimes|0\rangle \rightarrow s_{\mathbf{v}}^{-1/2}\sum_{i:v_i\neq 0}|i\rangle\otimes|v_i\rangle.$$

Using the index stored in the first register, conditionally rotate an ancilla register based on the ith entry of $\mathbf{v}$ and perform a second QRAM query to uncompute the second register such that the state of the system is

$$|\tilde{\mathbf{v}}\rangle = \frac{1}{\sqrt{s_{\mathbf{v}}}}\sum_{i:v_i\neq 0}|i\rangle\otimes\left(\sqrt{1-c_{\mathbf{v}}^2 v_i^2}|0\rangle + c_{\mathbf{v}} v_i|1\rangle\right),$$

where $c_{\mathbf{v}} \leq \min_i |v_i|^{-1}$. Conditioned on the ancilla qubit being in state $|1\rangle$, the first register is in state $|\mathbf{v}\rangle = \frac{\mathbf{v}}{||\mathbf{v}||}$. This state preparation procedure could potentially suffers from low probability of successfully projecting on the correct ancilla state when the vector is vastly dominated by a handful of large value entries. Fortunately, this is not the case in many realistic applications for Gaussian processes such as the datasets listed in Appendix C of Ref. [13]. It is also worth noting that even if the dataset is indeed only dominated by a constant number of large entries, under the moderate assumption that the predictive performance of the model is robust against minor element-wise perturbations in the input dataset, the state preparation can still succeed with a constant probability [29].

The second element necessary is a mechanism to estimate $\langle\mathbf{u}|\mathbf{v}\rangle$ for a given pair of real vectors $\mathbf{u}$ and $\mathbf{v}$. While the square of this value is accessible via a controlled-swap test, as discussed in [30], we require information about both the magnitude and sign of this inner product which is not accessible with such a test. In order to estimate the inner product, we instead apply a modified version of the state preparation procedure, where an additional ancilla qubit initially prepared in state $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$ is used to determine whether the target state is $|\mathbf{u}\rangle$ or $|\mathbf{v}\rangle$. This results in a joint state

$$\begin{aligned}|\Phi_{\mathbf{u},\mathbf{v}}\rangle =& \frac{1}{\sqrt{2s_{\mathbf{u}}}}\sum_{i:u_i\neq 0}|0\rangle|i\rangle\left(\sqrt{1-c_{\mathbf{u}}^2 u_i^2}|0\rangle + c_{\mathbf{u}} u_i|1\rangle\right)\\ &+ \frac{1}{\sqrt{2s_{\mathbf{v}}}}\sum_{i:v_i\neq 0}|1\rangle|i\rangle\left(\sqrt{1-c_{\mathbf{v}}^2 v_i^2}|0\rangle + c_{\mathbf{v}} v_i|1\rangle\right).\end{aligned}$$

Then the expectation value of the operator $X\otimes I\otimes|1\rangle\langle 1|$ is ${s_{\mathbf{u}}}^{-1/2}{s_{\mathbf{v}}}^{-1/2}c_{\mathbf{u}}c_{\mathbf{v}}\mathbf{u}^T\mathbf{v}$. As an alternative technique, the inner product can also be determined with the SWITCH test [31].

These two elements can be combined with the quantum linear systems algorithm to compute quantities of the form $\mathbf{u}^T A^{-1}\mathbf{v}$ as follows:

1. Initialise the system in state $|+\rangle_A|0\rangle_B|0\rangle_C|0\rangle_D$, where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$ and where A, B, C and D label distinct registers.

2. Controlled on register A being in state $|0\rangle$, prepare registers B and C in state $|\tilde{\mathbf{u}}\rangle$ such that the ancilla qubit is placed in register C with the remainder of the state in register B, and apply X to register D.

3. Controlled on register A being in state $|1\rangle$, prepare registers B and C in state $|\tilde{\mathbf{v}}\rangle$ such that the ancilla qubit is placed in register C with the remainder of the state in register B.

4. Controlled on both registers A and C being in state $|1\rangle$, apply the quantum linear systems algorithm to register B using register D as the ancilla used in the QLSA. A fifth register E is used for the phase estimation step in the QLSA, but since it is returned to the zero state, we do not explicitly include it in the description of the states after each step.

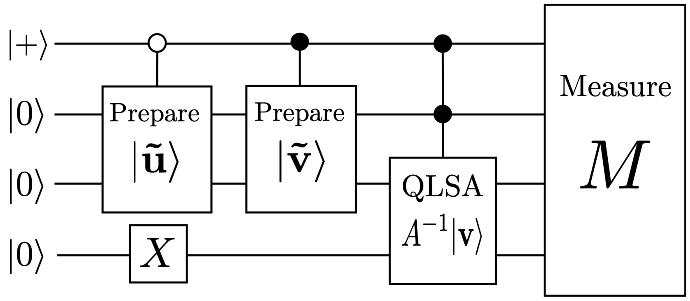
$|+\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
Prepare $|\tilde{\mathbf{u}}\rangle$
Prepare $|\tilde{\mathbf{v}}\rangle$
X
QLSA $A^{-1}|\mathbf{v}\rangle$
Measure M

FIG. 1: Circuit diagram for computing the form $\mathbf{u}^TA^{-1}\mathbf{v}$, where $M = X\otimes I\otimes|1\rangle\langle1|\otimes|1\rangle\langle1|$.

5. Measure the observable $M = X_AI_B|1\rangle\langle1|_C|1\rangle\langle1|_D$.

The measurement result is then a random variable with expectation value $cs_{\mathbf{u}}^{-1/2}s_{\mathbf{v}}^{-1/2}c_{\mathbf{u}}c_{\mathbf{v}}\mathbf{u}^TA^{-1}\mathbf{v}$. To see this, note that the state of the system after the fourth step is given by

$$\frac{1}{\sqrt{2s_{\mathbf{u}}}}|0\rangle_A\sum_{i:u_i\neq0}|i\rangle_B\left(\sqrt{1-c_{\mathbf{u}}^2u_i^2}|0\rangle_C+c_{\mathbf{u}}u_i|1\rangle_C\right)|1\rangle_D$$
$$+\frac{1}{\sqrt{2s_{\mathbf{v}}}}|1\rangle_A\sum_{i:v_i\neq0}c_{\mathbf{v}}\beta_i|\mu_i\rangle_B|1\rangle_C\left(\sqrt{1-\frac{c^2}{\lambda_i^2}}|0\rangle_D+\frac{c}{\lambda_i}|1\rangle_D\right)$$
$$+\frac{1}{\sqrt{2s_{\mathbf{v}}}}|1\rangle_A\sum_{i:v_i\neq0}\sqrt{1-c_{\mathbf{v}}^2v_i^2}|i\rangle_B|0\rangle_C|0\rangle_D$$

In the above $|\mu_i\rangle$ is taken to be the eigenvector of A corresponding to eigenvalue λ_i, and $\{\beta_i\}$ are taken to be the coordinates of $\mathbf{v}$ in the basis $\{|\mu_i\rangle\}$. Projecting this state onto $|1\rangle$ for registers C and D leads to the subnormalised state

$$\frac{1}{\sqrt{2s_{\mathbf{u}}}}|0\rangle_A\sum_{i:u_i\neq0}c_{\mathbf{u}}\gamma_i|\mu_i\rangle_B+\frac{1}{\sqrt{2s_{\mathbf{v}}}}|1\rangle_A\sum_{i:v_i\neq0}^{n}\frac{c}{\lambda_i}c_{\mathbf{v}}\beta_i|\mu_i\rangle_B,$$

where $\{\gamma_i\}$ are taken to be the coordinates of $\mathbf{u}$ in the basis given by $\{|\mu_i\rangle\}$. Thus, the expectation value for the measurement in the final step is

$$\sum_i\frac{1}{4}\left(\left(\frac{c_{\mathbf{u}}}{\sqrt{s_{\mathbf{u}}}}\gamma_i+\frac{c_{\mathbf{v}}c}{\sqrt{s_{\mathbf{v}}}}\frac{\beta_i}{\lambda_i}\right)^2-\left(\frac{c_{\mathbf{u}}}{\sqrt{s_{\mathbf{u}}}}\gamma_i-\frac{c_{\mathbf{v}}c}{\sqrt{s_{\mathbf{v}}}}\frac{\beta_i}{\lambda_i}\right)^2\right).$$

Since this is equal to $\frac{c_{\mathbf{u}}c_{\mathbf{v}}c}{\sqrt{s_{\mathbf{u}}s_{\mathbf{v}}}}\mathbf{u}^TA^{-1}\mathbf{v}$, the expectation value for the measurement in the final step, $\langle M\rangle$ must match this value. The estimation of $\langle M\rangle$ then involves sampling m repeated runs of the algorithm, resulting in a sampling variance that scales as m^{-1}. We illustrate the circuit diagram for this algorithm in FIG. 1.

The algorithm outlined above can be used to construct a quantum algorithm for approximating both the mean predictor and variance in GP regression, as follows:

- To approximate the mean predictor, $\mathbf{k}_*^T(K+\sigma_n^2I)^{-1}\mathbf{y} = \mathbf{y}^T(K+\sigma_n^2I)^{-1}\mathbf{k}_*$, we take $\mathbf{u}=\mathbf{y}$, $A = K+\sigma_n^2I$ and $\mathbf{v}=\mathbf{k}_*$. Since K is positive semi-definite the minimum eigenvalue of A is at least σ_n^2, and hence we take $c=\sigma_n^2$ in each run of the QLSA. This yields $\langle M\rangle = \frac{\sigma_n^2c_{\mathbf{k}_*}c_{\mathbf{y}}}{\sqrt{s_{\mathbf{k}_*}s_{\mathbf{y}}}}\mathbf{y}^T(K+\sigma_n^2I)^{-1}\mathbf{k}_*$, and hence
$$\bar{f}_* = \frac{\sqrt{s_{\mathbf{k}_*}s_{\mathbf{y}}}}{\sigma_n^2c_{\mathbf{k}_*}c_{\mathbf{y}}}\langle M\rangle. \tag{9}$$
Here $c_{\mathbf{k}_*}$ and $c_{\mathbf{y}}$ are taken to be the inverted maximum absolute values of the entries in $\mathbf{k}_*$ and $\mathbf{y}$, which we take to be constants. Hence the estimation variance in the computation of $\bar{f}_*$ will scale as $s_{\mathbf{k}_*}s_{\mathbf{y}}m^{-1}$. In the case of an s-sparse K, we have $s_{\mathbf{k}_*}\leq s$ since $\mathbf{k}_*$ reflects the same dependencies as K. While $\mathbf{y}$ will not in general be sparse, we can instead replace it in the estimation procedure with a sparse vector $\mathbf{y}'$ and still obtain a good approximation to $\bar{f}_*$, whenever the spectral norm of $K+\sigma_n^2I$ is bounded, which will essentially always be the case for GP regression as the eigenvalues of the covariance matrix do not typically scale up with the data size. This is because $(K+\sigma_n^2I)^{-1} = \sum_d(-1)^d(K+(\sigma_n^2-1)I)^d$, where we have used the Taylor series, $(I-S)^d = \sum_dS^d$, taking $S=-K+(1-\sigma_n^2)I$. The convergence of this series is guaranteed as long as the spectral norm of K is bounded and hence S can be normalised such that $\|S\|_*\leq1$. As such $(K+\sigma_n^2I)^{-1}$ can be approximated by a polynomial in $(K+(\sigma_n^2-1)I)$ of some fixed degree, which amounts to a matrix of constant sparsity. Hence $(K+\sigma_n^2I)^{-1}\mathbf{k}_*$ will be an approximately sparse vector, and its inner product with $\mathbf{y}$ can be well approximated by the inner product with a vector $\mathbf{y}'$ where the only non-zero entries correspond to the location of non-negligible entries of $(K+\sigma_n^2I)^{-1}\mathbf{k}_*$. Hence only a constant number of repetition of the algorithm is needed in order to achieve a fixed variance of estimation.
- To approximate the variance $\mathbb{V}[f_*]$ the same procedure is followed as for the mean predictor, except that $\mathbf{u}$ is taken to be $\mathbf{k}_*$ instead of $\mathbf{y}$. This yields $\langle M\rangle = \frac{\sigma_n^2c_{\mathbf{k}_*}^2}{s_{\mathbf{k}_*}}\mathbf{k}_*^T(K+\sigma_n^2I)^{-1}\mathbf{k}_*$, and hence
$$\mathbb{V}[f_*] = k(\mathbf{x}_*,\mathbf{x}_*)-\frac{s_{\mathbf{k}_*}}{\sigma_n^2c_{\mathbf{k}_*}^2}\langle M\rangle. \tag{10}$$
As with the mean predictor, $\langle M\rangle$ is measured on constant multiple independent runs of the algorithm to yield the fixed desired variance on the estimate.

The above shows how QLSA can be applied to computing two central objective quantities in Gaussian process regression problems. When $(K+\sigma_n^2I)$ is sparse and well-conditioned, this procedure achieves an exponential speed-up over classical Gaussian process regression.

V. DISCUSSION AND CONCLUSION

Gaussian Processes with sparse covariance matrices are of significant interests in many real world applications, particularly when the problem involves inference from large datasets [32]. For example, these sparsely constructed Gaussian processes are used to make a unified framework for robotic mapping [33]. In the field of pattern recognition, sparsely constructed Gaussian processes have been used to solve realistic action recognition problems [34]. A widely used technique to construct a sparse covariance matrix is setting the covariance function to zero beyond a certain distance between any two data points with a compactly supported function. Since each column or row of resultant matrix records the covariance function between a single data point and all data points, they only have a constant number of non-zero entries. By definition, this gives corresponds a sparse covariance matrix. This technique is known as covariance tapering, and has been proven to approximate the Matérn family of covariance functions with a small squared error [35]. An explicit example in geostatistics kriging where the dataset gives rise to a highly sparse covariance matrix is presented in [36]. In the above cases where the GPR computation only involves sparse covariance matrices, our proposed algorithm circumvents the major potential caveats of QLSA and an exponential advantage over its classical counter-part is attainable. For other applications where s scales linearly with n, our algorithm provides a polynomial speed-up over the best-known classical GPR algorithm, even though an exponential speed-up is not always guaranteed. Specifically, by applying a variant version of the QLSA, one can achieve a $\tilde{\mathcal{O}}(\sqrt{n})$ runtime [37].

In order to implement quantum GPR efficiently, the matrix $(K + \sigma_n^2 I)$ needs to be well-conditioned. The ratio of largest and smallest eigenvalue κ needs to stay low as n increases in order for the matrix to be robustly invertible. In classical GPR, conditioning is already a well-recognised issue. A general strategy to cope with the problem is to increase the noise variance $\sigma_n^2 I$ manually by a certain amount to dilute the ratio κ without severely affecting the statistical properties of the model. This increase in $\sigma_n^2 I$ can be seen as a small amount of noise (jitter) in the input signal. This technique is not new to the quantum GPR, and may be seen throughout the classical GP literature and mainstream implementations [18]. Therefore, for almost all practical purposes, we can assume the matrix is well-conditioned before applying the quantum algorithm. Moreover, when we apply our algorithm on a sparse kernel, we can employ the preconditioning method constructed in [22] to further suppress the growth of κ. In fact when K is sparse, under the realistic assumption that the magnitude of the entries in K does not scale with the dimensionality, the maximum eigenvalue of $(K + \sigma_n^2 I)$ will also be upper bounded by a constant. This is a consequence of the Gershgorin circle theorem [38] which can be expressed as

$$|\lambda - A_{ii}| \leq \sum_{j \neq i} |A_{ij}|, \tag{11}$$

where we set $A = (K + \sigma_n^2 I)$. Note that the minimum eigenvalue of A is lower bounded by σ_n^2. Likewise, we have the diagonal elements bounded by $A_{ii} \geq \sigma_n^2$ and the off-diagonal sum $\sum_{j \neq i} |A_{ij}|$ upper bounded by the sparsity of A scaled by the magnitude of its maximum entry. Hence Eq. 11 implies the condition number of A is bounded by a constant independent of n. As a result, under the sparse and bounded element kernel matrix assumption, conditioning does not provide a barrier to our proposed quantum GPR algorithm.

We have presented a novel procedure to apply the quantum algorithm for solving linear systems to Gaussian process regression modelling problems in supervised machine learning. By repeated sampling of the results of specific quantum measurements on the output states of QLSA, the mean predictor and the associated variance can be estimated with bounded error with potentially exponential speed-up over classical algorithms. Finally, we return to the discussion about the expected performance of our algorithm in terms of the four potential caveats raised by Ref. [17]. Providing access to a QRAM that stores input vectors, we have shown a technique to efficiently construct input states for the quantum algorithm. Regarding solution readout, we have constructed a procedure to estimate an inner product form that allows for efficient computation of the mean and variance predictor of GPR. Furthermore, we note that a large class of widely applied kernels in GPR give rise to sparse covariance matrices, and in such cases an exponential quantum speed-up is achievable. We have also argued that the Gershgorin circle theorem implies well-conditioning for sparse covariance matrices with bounded elements. Even in cases where the covariance matrix is necessarily dense, classical preconditioning techniques can still be applied and the quantum algorithm can still provide a polynomial advantage. Hence having addressed all the major potential caveats of QLSA [17], we have shown the quantum GPR algorithm to be a robust application with practical significance.

Acknowledgements— We thank Stephen Roberts for useful comments on the manuscript. JFF acknowledges support from the Air Force Office of Scientific Research under grant FA2386-15-1-4082 and Ministry of Education Singapore. This material is based on research supported by the Singapore National Research Foundation under NRF Award No. NRF-NRFF2013-01.

[1] R. P. Feynman, International journal of theoretical physics **21**, 467 (1982).

[2] P. W. Shor, SIAM journal on computing **26**, 1484 (1997).

[3] L. K. Grover, in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (ACM, 1996), pp. 212–219.

[4] E. Aïmeur, G. Brassard, and S. Gambs, in *Advances in Artificial Intelligence* (Springer, 2006), pp. 431–442.

[5] K. L. Pudenz and D. A. Lidar, Quantum information processing **12**, 2027 (2013).

[6] M. Schuld, I. Sinayskiy, and F. Petruccione, Contemporary Physics **56**, 172 (2015).

[7] M. Schuld, I. Sinayskiy, and F. Petruccione, in *PRICAI 2014: Trends in Artificial Intelligence* (Springer, 2014), pp. 208–220.

[8] S. Lloyd, M. Mohseni, and P. Rebentrost, arXiv preprint

arXiv:1307.0411 (2013).

[9] P. Rebentrost, M. Mohseni, and S. Lloyd, Physical Review Letters **113**, 130503 (2014).

[10] A. W. Harrow, A. Hassidim, and S. Lloyd, Physical Review Letters **103**, 150502 (2009).

[11] J. R. Shewchuk, Tech. Rep. CMU-CS-94-125, Department of Computer Science, Carnegie-Mellon University (1994).

[12] J. Lee, Y. Bahri, R. Novak, S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, arXiv preprint arXiv:1711.00165v1 (2017).

[13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning series)* (The MIT Press, 2005), ISBN 026218253X.

[14] J. Quiñonero-Candela and C. E. Rasmussen, The Journal of Machine Learning Research **6**, 1939 (2005).

[15] J. Hensman, N. Fusi, and N. D. Lawrence, in *Uncertainty in Artificial Intelligence* (2013).

[16] M. P. Deisenroth and J. W. Ng, arXiv preprint arXiv:1502.02843 (2015).

[17] S. Aaronson, Nature Physics **11**, 291 (2015).

[18] J. Bernardo, J. Berger, A. Dawid, A. Smith, et al., Bayesian statistics **6**, 475 (1998).

[19] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Communications in Mathematical Physics **270**, 359 (2007).

[20] V. Bužek, R. Derka, and S. Massar, Physical Review Letters **82**, 2207 (1999).

[21] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge university press, 2010).

[22] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, Physical Review Letters **110**, 250504 (2013).

[23] M. J. Grote and T. Huckle, SIAM Journal on Scientific Computing **18**, 838 (1997).

[24] E. Chow, SIAM Journal on Scientific Computing **21**, 1804 (2000).

[25] A. M. Childs, R. Kothari, and R. D. Somma, arXiv preprint arXiv:1511.02306 (2015).

[26] D. W. Berry, A. M. Childs, and R. Kothari, arXiv preprint arXiv:1501.01715 (2015).

[27] L. Wossnig, Z. Zhao, and A. Prakash, Phys. Rev. Lett. **120**, 050502 (2018).

[28] V. Giovannetti, S. Lloyd, and L. Maccone, Physical Review Letters **100**, 160501 (2008).

[29] Z. Zhao, V. Dunjko, J. K. Fitzsimons, P. Rebentrost, and J. F. Fitzsimons, arXiv preprint arXiv:1804.00281 (2018).

[30] L. Zhao, C. A. Pérez-Delgado, and J. F. Fitzsimons, arXiv preprint arXiv:1510.03742 (2015).

[31] P. Chamorro-Posada and J. C. Garcia-Escartin, arXiv preprint arXiv:1706.06564 (2017).

[32] A. Melkumyan and F. Ramos, in *IJCAI* (2009), vol. 9, pp. 1936–1942.

[33] S. Kim and J. Kim, in *Field and Service Robotics* (Springer, 2015), pp. 319–332.

[34] L. Liu, L. Shao, F. Zheng, and X. Li, Pattern Recognition **47**, 3819 (2014).

[35] R. Furrer, M. G. Genton, and D. Nychka, Journal of Computational and Graphical Statistics **15**, 502 (2006).

[36] R. P. Barry and R. Kelley Pace, Communications in Statistics-Simulation and Computation **26**, 619 (1997).

[37] L. Wossnig, Z. Zhao, and A. Prakash (2017), 1704.06174.

[38] R. S. Varga, *Geršgorin and his circles*, vol. 36 (Springer Science & Business Media, 2010).