

Quantum partial least squares regression algorithm for multiple correlation problem

Yan-Yan Hou(侯艳艳), Jian Li(李剑), Xiu-Bo Chen(陈秀波), and Yuan Tian(田源)
Citation: Chin. Phys. B, 2022, 31 (**3**): 030304. DOI: 10.1088/1674-1056/ac1b84
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Low-loss belief propagation decoder with Tanner graph in quantum error-correction codes

Dan-Dan Yan(颜丹丹), Xing-Kui Fan(范兴奎), Zhen-Yu Chen(陈祯羽), and Hong-Yang Ma(马鸿洋)
Chin. Phys. B, 2022, 31 (**1**): 010304. DOI: 10.1088/1674-1056/ac11cf

Semi-quantum private comparison protocol of size relation with *d*-dimensional GHZ states

Bing Wang(王冰), San-Qiu Liu(刘三秋), and Li-Hua Gong(龚黎华)
Chin. Phys. B, 2022, 31 (**1**): 010302. DOI: 10.1088/1674-1056/ac1413

Quantum computation and simulation with superconducting qubits

Kaiyong He(何楷泳), Xiao Geng(耿霄), Rutian Huang(黄汝田), Jianshe Liu(刘建设), and Wei Chen(陈炜)
Chin. Phys. B, 2021, 30 (**8**): 080304. DOI: 10.1088/1674-1056/ac16cf

Effects of initial states on the quantum correlations in the generalized Grover search algorithm

Zhen-Yu Chen(陈祯羽), Tian-Hui Qiu(邱田会), Wen-Bin Zhang(张文彬), and Hong-Yang Ma(马鸿洋)
Chin. Phys. B, 2021, 30 (**8**): 080303. DOI: 10.1088/1674-1056/ac05a9

Efficient self-testing system for quantum computations based on permutations

Shuquan Ma(马树泉), Changhua Zhu(朱畅华), Min Nie(聂敏), and Dongxiao Quan(权东晓)
Chin. Phys. B, 2021, 30 (**4**): 040305. DOI: 10.1088/1674-1056/abe29a

---

Quantum partial least squares regression algorithm for multiple correlation problem

Yan-Yan Hou(侯艳艳)[1,2,3], Jian Li(李剑)[1,†], Xiu-Bo Chen(陈秀波)[3,4], and Yuan Tian(田源)[1]

[1] *School of Artificial Intelligence, Beijing University of Post and Telecommunications, Beijing* 100876, *China*
[2] *College of Information Science and Engineering, Zaozhuang University, Zaozhuang* 277160, *China*
[3] *Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Post and Telecommunications, Beijing* 100876, *China*
[4] *GuiZhou University, Guizhou Provincial Key Laboratory of Public Big Data, Guiyang* 550025, *China*

(Received 27 April 2021; revised manuscript received 4 August 2021; accepted manuscript online 7 August 2021)

Partial least squares (PLS) regression is an important linear regression method that efficiently addresses the multiple correlation problem by combining principal component analysis and multiple regression. In this paper, we present a quantum partial least squares (QPLS) regression algorithm. To solve the high time complexity of the PLS regression, we design a quantum eigenvector search method to speed up principal components and regression parameters construction. Meanwhile, we give a density matrix product method to avoid multiple access to quantum random access memory (QRAM) during building residual matrices. The time and space complexities of the QPLS regression are logarithmic in the independent variable dimension n, the dependent variable dimension w, and the number of variables m. This algorithm achieves exponential speed-ups over the PLS regression on n, m, and w. In addition, the QPLS regression inspires us to explore more potential quantum machine learning applications in future works.

Keywords: quantum machine learning, partial least squares regression, eigenvalue decomposition

PACS: 03.67.Ac, 03.67.Lx **DOI:** 10.1088/1674-1056/ac1b84

1. Introduction

Linear regression is a statistical technique for analyzing and modeling the linear relationship between independent and dependent variables. Applications of linear regression occur in almost every field, including engineering, economics, chemical, and social sciences, etc. According to the number of independent variables, linear regression is divided into univariate linear regression and multiple linear regression. Actual problems are usually associated with various factors, and thus multiple linear regression has been used more widely than univariate linear regression.[1] However, multiple correlations between variables often seriously influence the construction of multiple linear regression models and damage the stability of models. Partial least squares (PLS) regression is a multiple linear regression algorithm related to principal component analysis, which builds the regression model by extracting the principal components reflecting the variation between independent and dependent variables.[2] Compared with other linear regression algorithms, the PLS regression can better solve the multiple correlation problem and be effectively modeled when the number of independent variables is less than the independent variable dimension. However, when the independent variable dimension is large, the PLS regression has high time complexity and becomes difficult to implement on classical computers.

With the development of quantum computing, quantum algorithms show better computing performance over the corresponding classical ones.[3] Superposition and entanglement of quantum states make quantum algorithms especially suitable for high-dimensional data computation in machine learning.[4] Recently, researchers proposed a series of quantum machine learning algorithms, such as classification,[5–11] clustering,[12–15] dimensionality reduction,[16–22,24] and neural networks.[25–29] These algorithms accelerate the development of quantum machine learning. Large-scale matrix computation in linear regression has high computational complexity, and various quantum regression algorithms have been proposed to solve this problem. Based on the quantum algorithm for solving systems of linear equations (HHL),[30] Wiebe[31] suggested a quantum regression algorithm from statistics. The time complexity of this algorithm is logarithmic in the data matrix dimension when the data matrix is sparse and well-conditioned. Schuld[32] established a quantum least-squares linear regression model that achieved an exponential acceleration in data size and significantly improved the dependency on the condition number. Wang[33] proposed an improved quantum linear regression algorithm. What is remarkable is that this algorithm can output regression parameters in classical form. In general, quantum regression algorithms have exponential speed-ups over the corresponding classical algorithms under certain conditions. However, the existing quantum regression algorithms are based on univariate linear regression models and can not solve the multiple correlation problem in regression.

†Corresponding author. E-mail: lijian@bupt.edu.cn

© 2022 Chinese Physical Society and IOP Publishing Ltd

In this paper, we design a quantum partial least squares (QPLS) regression algorithm for the multiple correlation problem. Our work has three significant contributions: (i) We propose the first quantum PLS regression algorithm to speed up the regression function construction. (ii) A quantum eigenvector search method is given to parallel search the specified eigenvector, which provides a novel idea for mapping high-dimensional data to low-dimensional space. (iii) A density matrix product method is designed to build residual matrix products, and this method can be a branch of other quantum algorithms to construct Hermitian matrix products.

The remainder of this paper is organized as follows. Section 2 reviews the PLS regression algorithm. Section 3 describes the QPLS regression algorithm and the related circuits. Section 4 analyzes the space and time complexities of the QPLS regression algorithm. Finally, we give a conclusion and discuss the future research directions.

2. Review of the PLS regression

In this section, we briefly review the PLS regression. The kernel work of the PLS regression is to build a linear regression function by projecting independent variables and dependent variables into the new feature space. Based on the linear regression function, we can predict the dependent variable for any independent variable. Let $X = \{x_1, \ldots, x_m\}$ be the set of the independent variables $x_i \in \mathbb{R}^n$ and $Y = \{y_1, \ldots, y_m\}$ be the set of the dependent variables $y_i \in \mathbb{R}^w$. We centralize and normalize X and Y, respectively, and get the independent variable matrix $E_0 \in \mathbb{R}^{n\times m}$ and the dependent variable matrix $F_0 \in \mathbb{R}^{n\times w}$. The PLS regression algorithm is an iterative process, and we firstly analyze the first iteration.

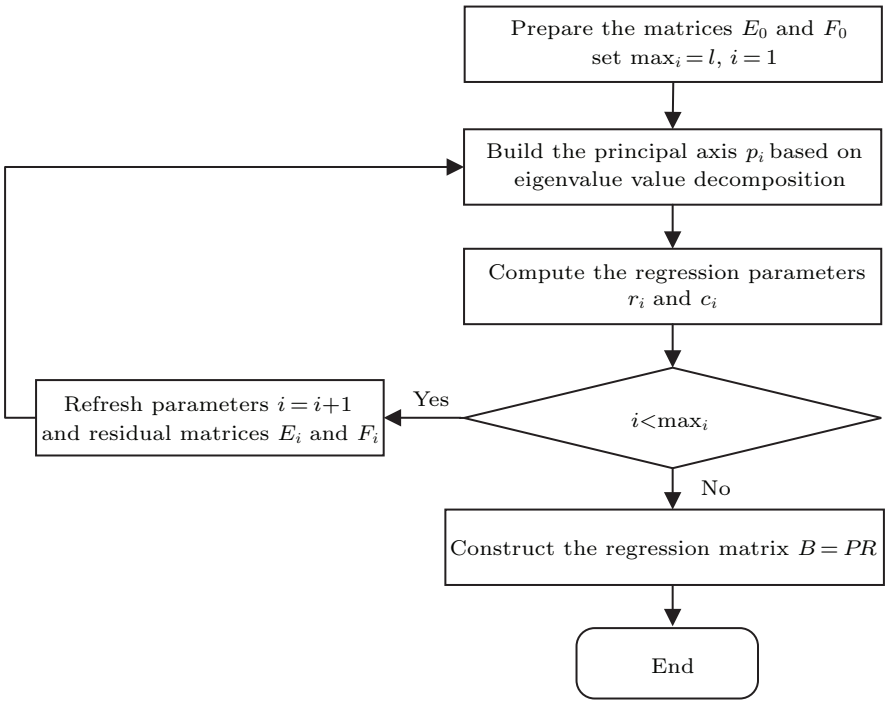


Fig. 1. The flow diagram of the PLS regression algorithm, i is the index of the current iteration, variable $\max_i$ stores the total number of iterations.

The first work is to find the principal axes (projection directions) p_1 and q_1 to create the principal components h_1 and g_1 such that these two principal components have maximum covariance, where $h_1 = E_0^{\mathrm{T}} p_1$ represents the linear combination of the columns of E_0 and $g_1 = F_0^{\mathrm{T}} q_1$ represents the linear combination of the columns of F_0. This work corresponds to the following optimization problem:

$$\max\langle E_0^{\mathrm{T}} p_1, F_0^{\mathrm{T}} q_1\rangle \quad \text{subject to}: \|p_1\| = 1,\ \|q_1\| = 1, \quad (1)$$

where $\langle,\rangle$ means the inner product operation, and $||\cdot||$ represents the l_2 norm. Lagrange multiplier method is a way to solve the extremum of multivariate function under a set of constraints. Introducing Lagrange multipliers α_1 and α_2, we transform Eq. (1) into the unconstrained optimization problem

$$L = p_1^{\mathrm{T}} E_0^{\mathrm{T}} F_0 q_1 - \alpha_1(p_1^{\mathrm{T}} p_1 - 1) - \alpha_2(q_1^{\mathrm{T}} q_1 - 1). \quad (2)$$

Take partial derivatives of L with respect to p_1, q_1, α_1, and α_2, equating them to 0 gives equations: $E_0^{\mathrm{T}} F_0 q_1 = 2\alpha_1 p_1$, $F_0^{\mathrm{T}} E_0 p_1 = 2\alpha_2 q_1$, $p_1^{\mathrm{T}} p_1 = 1$, and $q_1^{\mathrm{T}} q_1 = 1$. Based on the above four equations, we can deduce that

$$E_0^{\mathrm{T}} F_0 F_0^{\mathrm{T}} E_0 p_1 = \beta_1^2 p_1, \quad (3)$$

$$F_0^{\mathrm{T}} E_0 E_0^{\mathrm{T}} F_0 q_1 = \beta_2^2 q_1, \quad (4)$$

where $\beta_1 = \beta_2 = 2\alpha_1 = 2\alpha_2$, and $\beta_1 = \langle E_0^{\mathrm{T}} p_1, F_0^{\mathrm{T}} q_1\rangle$ corresponds to the target value of the optimization function. Once

β_1 gets the maximum, β_1^2 is the largest eigenvalue of the matrix $E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0$, and p_1 is the corresponding normalized eigenvector. Therefore, the principal axis p_1 can be obtained by performing eigenvalue decomposition on the matrix $E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0$. Subsequently, we compute the first principal component $h_1 = E_0p_1$ and construct the regression functions $E_0 = h_1c_1^{\mathrm{T}} + E_1$ and $F_0 = h_1r_1^{\mathrm{T}} + F_1$, where E_1 and F_1 are the residual matrices, and $c_1 = E_0^{\mathrm{T}}h_1/||h_1||^2$ and $r_1 = F_0^{\mathrm{T}}h_1/||h_1||^2$ are the regression parameters.

In the i-th iteration, we repeat the operations of the first iteration with the residual matrices E_{i-1} and F_{i-1} as the input, then get the principal axis p_i and the regression parameter r_i. After l iterations, we adopt all the principal axes p_i and the regression parameters r_i to construct the final regression function

$$F_0 = h_1r_1^{\mathrm{T}} + h_2r_2^{\mathrm{T}} + \cdots + h_lr_l^{\mathrm{T}} + F_l. \tag{5}$$

Define $H = \{h_1, h_2, \ldots, h_l\}$, $P = \{p_1, p_2, \ldots, p_l\}$, and $R = \{r_1, r_2, \ldots, r_l\}$, the regression function can be rewritten in matrix form as $F_0 = HR + F_l = E_0PR + F_l$. According to the truncation rule of principal component analysis, we can represent the regression function in low-rank form as $F_0 = E_0B$, where $B = PR^{\mathrm{T}}$ is the regression matrix. Figure 1 shows the flow diagram of the PLS regression algorithm.

Given a new independent variable x^*, we can predict the dependent variable y^* based on the prediction function $y^* = x^*B$. The prediction function can also be written in the vector form as $y^* = x^*\sum_{i=1}^{l} p_ir_i^{\mathrm{T}}$.

3. QPLS regression algorithm

In this section, we incorporate the design of the quantum eigenvector search method and the residual matrix product method into implementing the QPLS regression algorithm. Firstly, the quantum eigenvector search method is designed to compute the quantum state of the principal axis p_i. Secondly, the quantum state of the regression parameter r_i is constructed by combining the quantum eigenvector search method and the inner product operation. Thirdly, the density matrix product method is developed to build the residual matrix product $E_i^{\mathrm{T}}F_iF_i^{\mathrm{T}}E_i$. Finally, we create a quantum prediction function to predict the dependent variables for new independent variables. Now, we show the implementation process of the first iteration.

3.1. Quantum state preparation

QRAM[34] is a device that stores and accesses data in quantum superposition states. Kerenidis[35] proposed an improved QRAM to speed up data access exponentially, and we adopt this QRAM to prepare the input quantum state. Let $Z \in \mathbb{R}^{n\times w}$ denote an $n \times w$ matrix and $z_i = \{z_{i1}, z_{i2}, \ldots, z_{iw}\} \in \mathbb{R}^w$ be the i-th column of the matrix Z, where z_{ij} represents the $(i,\ j)$-th entry of the matrix Z. Let $|\varphi_{z_i}\rangle = \frac{1}{||z_i||}\sum_{j=1}^{w} z_{ij}|j\rangle$ be the quantum state of z_i and $|\varphi_Z\rangle = \frac{1}{\sqrt{N_Z}}\sum_{i=1}^{n}||z_i||i\rangle|\varphi_{z_i}\rangle = \frac{1}{\sqrt{N_Z}}\sum_{i=1}^{n}\sum_{j=1}^{w} z_{ij}|i\rangle|j\rangle$ be the quantum state of the matrix Z, where $N_Z = \sum_{i=1}^{n}||z_i||^2$ is a normalized constant. We define the unitary operations $U_{\mathcal{M}} : |i\rangle|0\rangle \to |i, z_i\rangle = \frac{1}{||z_i||}\sum_{j=1}^{n} z_{ij}|i\rangle|j\rangle$ and $U_{\mathcal{N}} : |0\rangle|j\rangle \to |Z_{\mathrm{F}}, j\rangle = \frac{1}{||Z||_{\mathrm{F}}}\sum_{i=1}^{w}||z_i||i\rangle|j\rangle$ to access quantum superposition states and the Frobenius norm of vectors, respectively, where $\|\cdot\|_{\mathrm{F}}$ represents the Frobenius norm. $|\varphi_Z\rangle$ can be accessed by performing $U_{\mathcal{N}}$ and $U_{\mathcal{M}}$ operations in turn.

3.2. Quantum eigenvector search method

In this subsection, we mainly design the quantum eigenvector search method to build the quantum state of the principal axis p_1. According to Eq. (3), the principal axis p_1 is the normalized eigenvector corresponding to the largest eigenvalue of the matrix $E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0$. We first prepare a density matrix to denote $E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0$. Let $|\phi\rangle = \sum_{j=1}^{n}\sum_{k=1}^{w}\phi_{jk}|j\rangle|k\rangle$ denote the quantum state of the matrix $E_0^{\mathrm{T}}F_0$, where ϕ_{jk} corresponds to the (i,j)-th normalized entry of $E_0^{\mathrm{T}}F_0$. $|\phi\rangle$ can be rewritten in Schmidt decomposition form as $|\phi\rangle = \sum_{i=1}^{r_m}\sigma_i|u_i\rangle|v_i\rangle$, where r_m is the rank of $E_0^{\mathrm{T}}F_0$, σ_i is the i-th normalized singular value of $E_0^{\mathrm{T}}F_0$, and $|u_i\rangle$ and $|v_i\rangle$ are the corresponding left and right singular vectors, respectively. Let $\rho_\phi = \sum_{i,i'=1}^{r_m}\sigma_i\sigma_{i'}|u_i\rangle\langle u_{i'}||v_i\rangle\langle v_{i'}|$ be the density matrix of $|\phi\rangle$. Take partial trace over the second register of ρ_ϕ, then we obtain the reduced density matrix $\rho_0 = \sum_{i,i'=1}^{r_m}\sigma_i\sigma_{i'}|u_i\rangle\langle u_{i'}|$, which is the density matrix of $E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0$.

We design the quantum eigenvector search method to search the eigenvector $|u_1\rangle$ corresponding to the largest eigenvalue $\lambda_1 = \sigma_1^2$ of ρ_0. The first work is to perform eigenvalue decomposition on ρ_0. The density matrix exponentiation method[23] allows us to adopt multiple copies of ρ_0 to construct the unitary operation $\mathrm{e}^{\mathrm{i}\rho_0 t}$. We set the state $|\varphi^1\rangle = |\phi\rangle_{1,2}|0\rangle_3 = \sum_{i=1}^{r_m}\sigma_i|u_i\rangle_1|v_i\rangle_2|0\rangle_3$ as the input, where registers 1 and 2 store $|\phi\rangle$, and register 3 stores the estimated eigenvalues, then perform phase estimation on $|\varphi^1\rangle$ with $\mathrm{e}^{\mathrm{i}\rho_0 t}$ as the controlled unitary. Finally, we get the state

$$|\varphi^2\rangle = \sum_{i=1}^{r_m}\sigma_i|u_i\rangle_1|v_i\rangle_2|\widetilde{\lambda_i}\rangle_3, \tag{6}$$

where $\widetilde{\lambda_i}$ is the estimate of λ_i, and $|u_i\rangle$ is the eigenvector corresponding to the first principal axis p_1. $|\varphi^2\rangle$ can be viewed as the superposition state of multiple eigenvectors $|u_i\rangle$, and $|u_1\rangle$ is the quantum state of the principal axis p_1.

The second work is to extract the eigenvector $|u_1\rangle$ from $|\varphi^2\rangle$. QPCA[36] is a quantum principal component analysis method for quantum state tomography. The eigenvectors and eigenvalues can be estimated by measuring the eigenvalue register, and this procedure will be highly efficient when the matrix is low rank. However, the probability of the quantum state

of the eigenvalue determines the number of the measurements, so this method would lose logarithmic speed-up when the data set corresponds to an exponentially sizeable high-dimensional matrix. Yu[37] presented another quantum principal component analysis method, which could project high-dimensional data into low-dimensional space, but this method relies on the eigenvalues sorted in descending order. Unfortunately, eigenvalues are stored disorderly in a quantum superposition state, and it is not easy to obtain the ordered eigenvalues.[38] Here, we adopt quantum comparison and amplitude estimation to design a quantum binary search method. This method can extract the eigenvector $|u_1\rangle$ without the ordered eigenvalues and quantum state tomography. The specific process is as follows.

(1) Initialize comparison times $l_s = \log(1/\varepsilon)$, where ε is the error of the estimated eigenvalues. Since all $\widetilde{\lambda}_i$ are in the range $(0,1]$, we initialize the lower bound $\tau_{\min} = 0$, the upper bound $\tau_{\max} = 1$, and the comparison threshold $\tau = (\tau_{\min} + \tau_{\max})/2$.

(2) Compare the estimated eigenvalue $|\widetilde{\lambda}_i\rangle$ with the comparison threshold $|\tau\rangle$. We first prepare the initial state $|\varphi^2\rangle|\tau\rangle_4|0\rangle_5|0\rangle_6 = \sum_{i=1}^{r_m} \sigma_i|u_i\rangle_1|v_i\rangle_2|\widetilde{\lambda}_i\rangle_3|\tau\rangle_4|0\rangle_5|0\rangle_6$ as the input, where register 4 stores the comparison threshold $|\tau\rangle$ and registers 5 and 6 store the comparison result. Let R_1 represent the comparison operation, which obeys: If $|\widetilde{\lambda}_i\rangle$ is less than $|\tau\rangle$, set register 6 to $|0\rangle$; otherwise, set register 6 to $|1\rangle$. After performing the comparison operation R_1, we get the state

$$\begin{aligned}|\varphi^3\rangle \;=\; & \sum_{\widetilde{\lambda}_i \geq \tau} \sigma_i|u_i\rangle_1|v_i\rangle_2|\widetilde{\lambda}_i\rangle_3|\tau\rangle_4|f_i\rangle_5|1\rangle_6 \\ & + \sum_{\widetilde{\lambda}_i < \tau} \sigma_i|u_i\rangle_1|v_i\rangle_2|\widetilde{\lambda}_i\rangle_3|\tau\rangle_4|f_i\rangle_5|0\rangle_6. \end{aligned} \tag{7}$$

$\sum_{\widetilde{\lambda}_i \geq \tau} \sigma_i|u_i\rangle_1|v_i\rangle_2|\widetilde{\lambda}_i\rangle_3|\tau\rangle_4|f_1\rangle_5|1\rangle_6$ is the projection of the state $|\varphi^3\rangle$ onto the space spanned by the specified eigenvectors, whose corresponding eigenvalues are greater than the threshold τ.

Figure 2 shows the circuit for a single qubit comparison operation. In this circuit, registers 3 and 4 store $|\widetilde{\lambda}_i\rangle$ and $|\tau\rangle$, respectively; registers 5 and 6 store the comparison result. If $|\widetilde{\lambda}_i\rangle > |\tau\rangle$, set registers 5 and 6 to $|1\rangle|1\rangle$; if $|\widetilde{\lambda}_i\rangle < |\tau\rangle$, set them to $|0\rangle|0\rangle$; otherwise, set them to $|0\rangle|1\rangle$. Figure 3 shows the circuit for the n qubits comparison operation. Similar to the classical comparison circuit, the comparison circuit for n qubits is implemented by n single qubit comparison circuits cascading. Let $\{a_n, a_{n-1}, \ldots, a_2, a_1\}$ and $\{b_n, b_{n-1}, \ldots, b_2, b_1\}$ represent the qubit sequences of registers 3 and 4, respectively. This circuit compares the qubit pair (a_i, b_i) in order from high-order qubit to low-order qubit and passes the comparison result of each qubit pair to the lower-order qubit comparisons. Register 5 contains the qubit sequences $\{c_n, c_{n-1}, \ldots, c_2, c_1\}$ and $\{d_{n-1}, \ldots, d_2\}$), which are used to store the comparison results of each qubits pair. Finally, we get the final comparison result from register 6 (the qubit d_1). Assume $|\widetilde{\lambda}_i\rangle$ and $|\tau\rangle$ are stored in registers 3 and 4, respectively, we perform the comparison operation R_1 and get the final comparison result: if $|\widetilde{\lambda}_i\rangle \geq |\tau\rangle$, register 6 is $|1\rangle$; otherwise, register 6 is $|0\rangle$. For the specific design method, refer to Ref. [39].

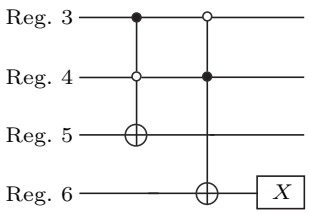


Fig. 2. Comparison circuit on a single qubit. Reg. 3, Reg. 4, Reg. 5, and Reg. 6 represent registers 3, 4, 5, and 6, respectively. Reg. 3 and Reg. 4 store $|\widetilde{\lambda}_i\rangle$ and $|\tau\rangle$, respectively; Reg. 5 and Reg. 6 store the comparison results.

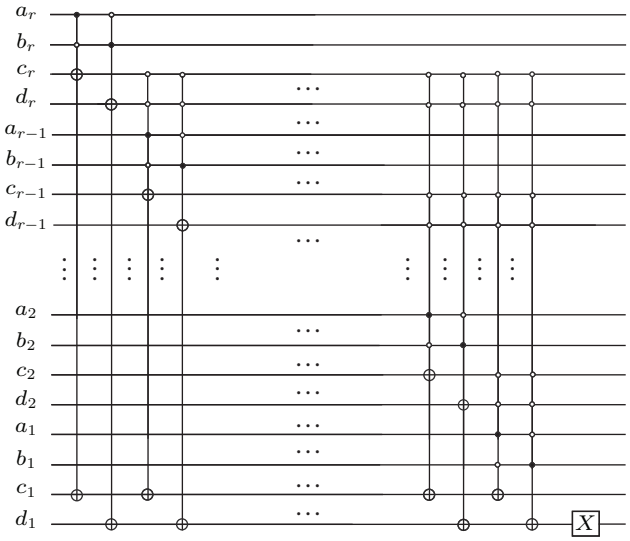


Fig. 3. Comparison circuit on n qubits. $\{a_n, a_{n-1}, \ldots, a_2, a_1\}$ and $\{b_n, b_{n-1}, \ldots, b_2, b_1\}$ represent the qubit sequences of register 3 and register 4, respectively. $\{c_n, c_{n-1}, \ldots, c_2, c_1\}$ and $\{d_n, d_{n-1}, \ldots, d_2\}$ represent the qubit sequence of register 5, and the qubit d_1 represents register 6. All qubit sequences are arranged in descending order. The circuit compares each qubit pair (a_i, b_i) in order from high-order qubit to low-order qubit and passes the comparison result of each qubit pair to the lower-order qubit comparison. The output of d_1 represents the final comparison result.

(3) Rewrite Eq. (7) as $|\varphi^3\rangle = \sin(\theta_1)|\varphi_1^3\rangle_{123456} + \cos(\theta_1)|\varphi_0^3\rangle_{123456}$, where $\sin(\theta_1)|\varphi_1^3\rangle_{123456}$ and $\cos(\theta_1)|\varphi_0^3\rangle_{123456}$ correspond to register 6 being $|1\rangle$ and $|0\rangle$, respectively. Let $R' = (2|\varphi_0^3\rangle\langle\varphi_0^3| - I)$ be the reflection operation through the space spanned by the vector $|\varphi_0^3\rangle$ and $R_1 R' R_1^\dagger = (2|\varphi^3\rangle\langle\varphi^3| - I)$ be the reflection operation through the space spanned by the vector $|\varphi^3\rangle$, where $R_1^\dagger$ is the inverse operation of R_1 and the subscript 123456 is dropped for the ease of exposition. Let G_1 be the unitary operation $R_1 R' R_1^\dagger R'$, which obeys

$$\begin{cases} G_1|\varphi_0^3\rangle = \cos(2\theta_1)|\varphi_0^3\rangle + \sin(2\theta_1)|\varphi_1^3\rangle, \\ G_1|\varphi_1^3\rangle = -\sin(2\theta_1)|\varphi_0^3\rangle + \cos(2\theta_1)|\varphi_1^3\rangle. \end{cases} \tag{8}$$

G_1 can be further written in the matrix form as

$$G_1 = \begin{bmatrix} \cos(2\theta_1), & -\sin(2\theta_1) \\ \sin(2\theta_1), & \cos(2\theta_1) \end{bmatrix}. \tag{9}$$

In the basis $\{|\varphi_0^3\rangle, |\varphi_1^3\rangle\}$, the eigenvalues of G_1 are $\lambda_\pm = e^{\pm 2i\theta_1}$, and the corresponding eigenvectors are $|\varphi_\pm^3\rangle = \frac{1}{\sqrt{2}}(|\varphi_1^3\rangle \mp i|\varphi_0^3\rangle)$. $|\varphi^3\rangle$ can be further written in eigenvalue decomposition form as $|\varphi^3\rangle = \frac{1}{\sqrt{2}}(e^{i\theta_1}|\varphi_+^3\rangle + e^{-i\theta_1}|\varphi_-^3\rangle)$.

(4) Set the state $|\varphi^3\rangle|0\rangle_7$ as the input, perform amplitude estimation with $e^{iG_1 t}$ as the controlled unitary, and get the state

$$|\varphi^4\rangle = \frac{1}{\sqrt{2}}(e^{i\theta_1}|\varphi_+^3\rangle|2\widetilde{\theta}_1\rangle_7 + e^{-i\theta_1}|\varphi_-^3\rangle|2\pi - 2\widetilde{\theta}_1\rangle_7), \quad (10)$$

where $\widetilde{\theta}_1$ is the estimated value of θ_1, then, measure register 7 and get $2\widetilde{\theta}_1$, where $\widetilde{\theta}_1$ corresponds to the probability of $\widetilde{\lambda}_i \geq \tau$. If $\widetilde{\theta}_1$ is 0, update $\tau_{\max} = \tau$ and $\tau = (\tau_{\max} + \tau_{\min})/2$ in turn; otherwise, set $\tau_{\min} = \tau$ and $\tau = (\tau_{\max} + \tau_{\min})/2$ in turn. Repeat steps (1) to (4) with $|\varphi^4\rangle$ as the input until completing l_s iterations and get the state

$$\begin{aligned}|\varphi^5\rangle = {} & \sigma_1|u_1\rangle_1|v_1\rangle_2|\widetilde{\lambda}_1\rangle_3|\tau\rangle_4|f_1\rangle_5|1\rangle_6 \\ & + \sum_{i\in S}\sigma_i|u_i\rangle_1|v_i\rangle_2|\widetilde{\lambda}_i\rangle_3|\tau\rangle_4|f_i\rangle_5|0\rangle_6, \end{aligned} \quad (11)$$

where $|u_1\rangle$ is the quantum state of p_1, S is the union of the eigenvalues satisfying the condition $\widetilde{\lambda}_i \neq \widetilde{\lambda}_1$.

(5) Measure register 6 into $|1\rangle$, and remove registers 2, 3, 4, and 5. Finally, we get the state $|u_1\rangle$ in register 1, which corresponds to the quantum state of the principal axis $|u_1\rangle$. It is worth mentioning that amplitude amplification can improve the probability of measurement success quadratically. The quantum eigenvector search method is described in Algorithm 1.

Algorithm 1 Quantum eigenvector search method

Input: the quantum state $|\phi\rangle$ for $E_0^{\mathrm{T}}F_0$, the reduced density matrix ρ_0 for $E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0$, the iteration times l_s.
Output: the quantum state $|u_1\rangle$ for the principal axis p_1.
1: Perform phase estimation on $|\phi\rangle$ and get the eigenvalue decomposition of $|\varphi^2\rangle$.
2: Initialize $i = 1$, $\tau_{\min} = 0$, $\tau_{\max} = 1$, and $\tau = (\tau_{\min} + \tau_{\max})/2$.
3: **while** $i \leq l_s$ **do**
4: Perform comparison operation R_1 on $|\varphi^2\rangle$ and $|\tau\rangle$ and get the comparison result in $|\varphi^3\rangle$.
5: Act amplitude estimation on $|\varphi^3\rangle$ to get θ representing the probability of $\widetilde{\lambda}_i < \tau$.
6: **If** $\theta < 0$ **then**
7: Update $\tau_{\max} = \tau$, and $\tau = (\tau_{\max} + \tau_{\min})/2$ in turn.
8: **else**
9: Update $\tau_{\min} = \tau$, and $\tau = (\tau_{\max} + \tau_{\min})/2$ in turn.
10: $i = i + 1$.
11: Measure register 6 with $|1\rangle$ to get $|u_1\rangle$ corresponding to the principal axis p_1.

The circuit of the quantum eigenvector search method is shown in Fig. 4, including phase estimation, comparison operation, and amplitude estimation.

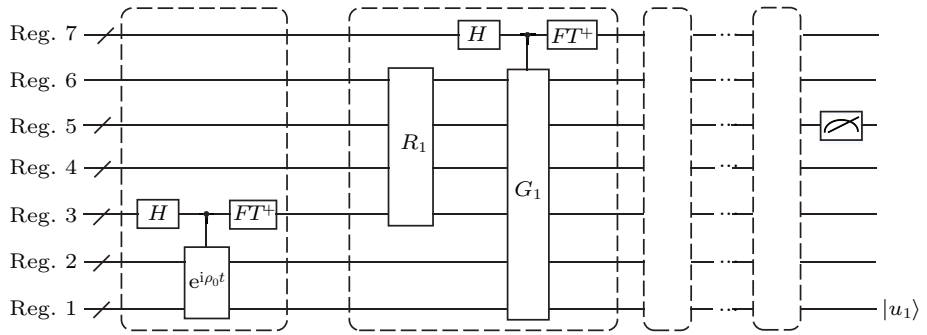


Fig. 4. The circuit of the quantum eigenvector search method. Reg. 1–Reg. 7 represent registers 1–7, respectively. The circuit in the first dotted box represents phase estimation, where H means Hadamard transform; $e^{i\rho_0 t}$ represents the unitary operation controlled by $e^{i\rho_0 t}$; FT^+ denotes the inverse operation of quantum Fourier transform. The circuit in the second dotted box represents the quantum binary search operation, R_1 denotes comparison operation, and the combination of unitary operations H, G_1, and FT^+ corresponds to amplitude estimation. The rest dotted boxes represent the repeated quantum binary search operations.

3.3. Construct regression parameters

In this subsection, we design a quantum method to construct the quantum state of regression parameter r_1 according to equation $r_1 = F_0^{\mathrm{T}}h_1/||h_1||^2$. In the first iteration, we first reformulate $E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0p_1 = \beta_1^2 p_1$ to facilitate the construction of the quantum state of r_1. Firstly, multiply both sides of $E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0p_1 = \beta_1^2 p_1$ by E_0, and get

$$E_0E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0p_1 = \beta_1^2 E_0p_1. \quad (12)$$

According to the condition $h_1 = E_0p_1$, Eq. (12) can be rewritten as

$$E_0E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}h_1 = \beta_1^2 h_1, \quad (13)$$

where h_1 is the eigenvector corresponding to the largest eigenvalue β_1^2 of the matrix $E_0E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}$. Multiply both sides of Eq. (13) by F_0^{T}, then

$$F_0^{\mathrm{T}}E_0E_0^{\mathrm{T}}F_0r_1' = \beta_1^2 r_1', \tag{14}$$

where $r_1' = {F_0}^{\mathrm{T}}h_1$ is the eigenvector corresponding to the largest eigenvalue β_1^2 of the matrix product $F_0^{\mathrm{T}}E_0E_0^{\mathrm{T}}F_0$. Since $F_0^{\mathrm{T}}E_0E_0^{\mathrm{T}}F_0$ is the transpose of $E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0$, we still adopt the quantum eigenvector search method to construct the quantum state $|\varphi^5\rangle' = \sigma_1|u_1\rangle_1|v_1\rangle_2|\widetilde{\lambda}_1\rangle_3|\tau\rangle_4|f_1\rangle_5|1\rangle_6 + \sum_{i\in S}\sigma_i|u_i\rangle_1|v_i\rangle_2|\widetilde{\lambda}_i\rangle_3|\tau\rangle_4|f_i\rangle_5|0\rangle_6$ (the copy of $|\varphi^5\rangle$). Different from step (5) in Subsection 3.2, we remove registers 1, 3, 4, and 5 after measuring register 6 into $|1\rangle$. Finally, we get $|v_1\rangle$, which corresponds to the quantum state of $r_1' = {F_0}^{\mathrm{T}}h_1$.

To further obtain the quantum state of the regression parameter $r_1 = r_1'/||h_1||^2$, we adopt the Bell-basis measurement method[40] to compute the constant $||h_1||^2 = \langle E_0p_1, E_0p_1\rangle$. Let $|\phi_{E_0}\rangle_{AB} = \sum_{i=1}^{m}\sum_{j=1}^{n}e_{ij}|i\rangle_A|j\rangle_B$ be the quantum state of the matrix E_0, where e_{ij} represents the normalized entry of the matrix E_0, registers A and B store the indices of the matrix E_0. Let $|u_1\rangle_C = \sum_{j=1}^{n}u_{1j}|j\rangle_C$ be the quantum state of the principal axis p_1. Figure 5 shows the circuit of computing $||h_1||^2$. The circuit mainly includes two models.

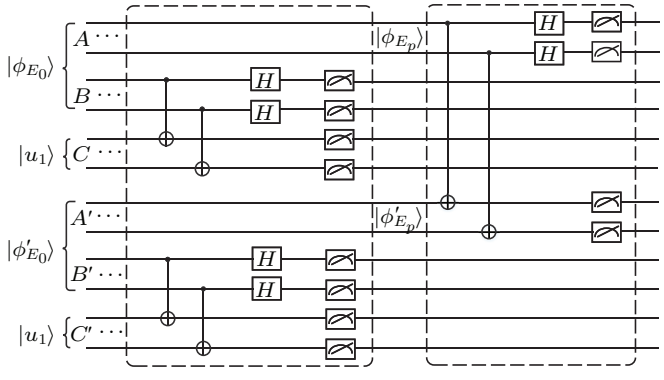


Fig. 5. Circuit of computing $||h_1||^2$. $|\phi_{E_0}\rangle$ and $|u_1\rangle$ represent the quantum states of the matrix E_0 and the principal axis p_1, respectively. Registers A and B store the quantum state $|\phi_{E_0}\rangle$, and register C stores the quantum state $|u_1\rangle$. Registers A' and B' store the quantum state $|\phi'_{E_0}\rangle$, the copy of $|\phi_{E_0}\rangle$. Register C' stores the quantum state $|u_1'\rangle$, the copy of $|u_1\rangle$. $|\phi_{Ep}\rangle$ and $|\phi'_{Ep}\rangle$ are the output of the first circuit model. Two dashed boxes represent two models of the circuit, respectively. The classical post-processing is omitted in this circuit.

(1) The first circuit model is mainly used to build the quantum states $|\phi_{Ep}\rangle$ and $|\phi'_{Ep}\rangle$, which respectively represent the quantum states of the vector E_0p_1 and its copy. Firstly, CNOT operations are applied on the corresponding qubit pairs of registers B and C, followed by H operations on all qubits of register B. Subsequently, every qubit of registers B and C is measured into $|0\rangle$. The output of register A corresponds to $|\phi_{Ep}\rangle$. Similarly, $|\phi'_{E_0}\rangle_{A'B'}$ and $|u_1'\rangle_{C'}$ are used to compute $|\phi'_{Ep}\rangle_{A'}$, where $|\phi'_{E_0}\rangle_{A'B'}$ and $|u_1'\rangle_{C'}$ are the copies of $|\phi_{E_0}\rangle_{AB}$ and $|u_1\rangle_C$, respectively, and A', B', and C' represent registers A', B', and C'.

(2) The second circuit model is responsible for computing the inner product $\langle\phi_{Ep}|\phi'_{Ep}\rangle$. Firstly, CNOT operations are performed on the corresponding qubit pairs of registers A, and A', followed by H operations on the qubits of register A. Subsequently, all qubit pairs of registers A and A' are measured. Let $\boldsymbol{p} = (p_{00}^1, p_{01}^1, p_{10}^1, p_{11}^1)\otimes\cdots\otimes(p_{00}^\gamma, p_{01}^\gamma, p_{10}^\gamma, p_{11}^\gamma)$ be the probability of the measurement result and $\boldsymbol{c} = (1,1,1,-1)^{\otimes\gamma}$ be the post-processing vector, where p_j^i represents the probability that the ith qubit pair is j and $\gamma = \log(m)$ is the number of qubits of register A. Finally, we adopt the classical post-processing to get $||h_1||^2 = \boldsymbol{p}\cdot\boldsymbol{c}$.

3.4. Build residual matrix product

According to the simple PLS regression (SIMPLS),[41] the principal component not only represents the independent variable set but has a strong explanation for the dependent variable set. Therefore, all the matrices F_i equals the matrix F_0, and there is no need to update the residual matrix F_i in each iteration. The PLS regression algorithm adopts multiple principal axes p_i and regression parameters r_i to build the regression matrix B. We prepare the residual matrices E_i and F_i in the i-th iteration, then perform eigenvalue decomposition on the matrix product $E_i^{\mathrm{T}}F_iF_i^{\mathrm{T}}E_i$ in the subsequent iteration. Since the residual matrices E_i and F_i are stored in QRAM, we need to multiple access QRAM to build the residual matrix product $E_i^{\mathrm{T}}F_iF_i^{\mathrm{T}}E_i$. Here, we reformulate the original residual matrices constructing method and design a density matrix product method to build $E_i^{\mathrm{T}}F_iF_i^{\mathrm{T}}E_i$. The advantage of this method is that each residual matrix can be direct as the input for the next iteration without access operations.

In the first iteration, the main work is to build the residual matrix product $E_1^{\mathrm{T}}F_1F_1^{\mathrm{T}}E_1$. According to the conditions $h_1 = E_0p_1$ and $c_1 = E_0^{\mathrm{T}}h_1/||h_1||^2$, convert $E_1 = E_0 - h_1c_1^{\mathrm{T}}$ into $E_1 = E_0(I - p_1{p_1}^{\mathrm{T}}E_0^{\mathrm{T}}E_0/||E_0p_1||)$, then

$$\begin{aligned} E_1^{\mathrm{T}}F_1F_1^{\mathrm{T}}E_1 \;=\; & \left(I - \frac{E_0^{\mathrm{T}}E_0p_1{p_1}^{\mathrm{T}}}{||E_0p_1||^2}\right)E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0 \\ & \times\left(I - \frac{E_0^{\mathrm{T}}E_0p_1{p_1}^{\mathrm{T}}}{||E_0p_1||^2}\right)^{\mathrm{T}}. \end{aligned} \tag{15}$$

The essential steps of the quantum density matrix product method are described as follows.

(1) Prepare the quantum state $|\delta\rangle = \sum_{i=1}^{n}\sum_{j=1}^{n}\delta_{ij}|i\rangle|j\rangle$ for the matrix $U_1 = E_0^{\mathrm{T}}E_0p_1{p_1}^{\mathrm{T}}/||E_0p_1||^2$. $|\delta\rangle$ can be written in Gram–Schmidt decomposition form as $|\delta\rangle = \sum_{k=1}^{r_n}\sigma_k^1|u_k^1\rangle|v_k^1\rangle$, where σ_k^1 is the k-th singular value of U_1 in descending order, $|u_k^1\rangle$ and $|v_k^1\rangle$ are the quantum states of the corresponding left and right singular vectors. Let $\rho_\delta = \sum_{k,k'=1}^{r_n}\sigma_i^1\sigma_{k'}^1|u_k^1\rangle\langle u_{k'}^1||v_k^1\rangle\langle v_{i'}^1|$ be the density matrix of $|\delta\rangle$. We

take the partial trace of ρ_δ over the second register, then get the reduced density matrix

$$\rho_{U_1} = \sum_{k,k'=1}^{r_n} \sigma_k^1 \sigma_{k'}^1 |u_k^1\rangle\langle u_{k'}^1|, \tag{16}$$

where $|u_k^1\rangle$ and $|u_{k'}^1\rangle$ correspond to the eigenvectors of the density matrix ρ_{U_1}. As defined in Subsection 3.2, the reduced density matrix $\rho_0 = \sum_{k,k'=1}^{n}\sum_{j=1}^{w} \phi_{kj}\phi_{k'j}|k\rangle\langle k'|$ is the quantum state of $E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0$. We rewrite ρ_0 in ρ_{U_1} eigenvalue decomposition form as

$$\rho_0 = \sum_{k,k'=1}^{r_n} \eta_k^1 \eta_{k'}^1 |u_k^1\rangle\langle u_{k'}^1|, \tag{17}$$

where η_k^1 and $\eta_{k'}^1$ are the projections of ρ_0 onto $|u_k^1\rangle$ and $|u_{k'}^1\rangle$, respectively.

(2) Perform phase estimation on $\rho_0|0\rangle\langle 0|$ with $\mathrm{e}^{\mathrm{i}\rho_{U_1}t}$ as the controlled unitary, then get

$$\rho_0' = \sum_{k=1}^{r_n} (\eta_k^1)^2 |u_k^1\rangle\langle u_k^1||\widetilde{\lambda}_k^1\rangle\langle\widetilde{\lambda}_k^1|, \tag{18}$$

where $\widetilde{\lambda}_k^1 = (\sigma_k^1)^2$ represents the estimated eigenvalue of ρ_{U_1}.

(3) Add an ancilla register with $|0\rangle\langle 0|$ and rotate it conditional on the second register, then obtain

$$\rho_0'' = \sum_{k=1}^{r_n} (\eta_k^1)^2 |u_k^1\rangle\langle u_k^1||\widetilde{\lambda}_k^1\rangle\langle\widetilde{\lambda}_k^1||\psi_{\lambda_k^1}\rangle\langle\psi_{\lambda_k^1}|, \tag{19}$$

where

$$|\psi_{\lambda_k^1}\rangle = \sqrt{1-\left(\frac{(1-\widetilde{\lambda}_k^1)}{\max_{k\in r_n}(1-\widetilde{\lambda}_k^1)}\right)^2}|1\rangle + \frac{(1-\widetilde{\lambda}_k^1)}{\max_{k\in r_n}(1-\widetilde{\lambda}_k^1)}|0\rangle.$$

(4) Perform inverse phase estimation and remove the second register. Subsequently, measure the ancilla register into $|0\rangle$, and get

$$\rho_1 = \frac{1}{\sqrt{N_1}} \sum_{k=1}^{r_n} \frac{(\eta_k^1)^2(1-\widetilde{\lambda}_k^1)}{\max_{k\in r_n}(1-\widetilde{\lambda}_k^1)} |u_k^1\rangle\langle u_k^1|, \tag{20}$$

where N_1 is a normalized constant. The density matrix ρ_1 corresponds to the quantum state of the residual matrix product $E_1^{\mathrm{T}}F_1F_1^{\mathrm{T}}E_1$.

In the second iteration, we build the residual matrix product

$$E_2^{\mathrm{T}}F_2F_2^{\mathrm{T}}E_2 = \left(I - \frac{E_1^{\mathrm{T}}E_1p_2p_2^{\mathrm{T}}}{||E_1p_2||^2}\right)E_1^{\mathrm{T}}F_1F_1^{\mathrm{T}}E_1 \times \left(I - \frac{E_1^{\mathrm{T}}E_1p_2p_2^{\mathrm{T}}}{||E_1p_2||^2}\right)^{\mathrm{T}}. \tag{21}$$

Let ρ_{U_2} represent the matrix $U_2U_2^{\mathrm{T}}$, where $U_2 = \frac{E_1^{\mathrm{T}}E_1p_2p_2^{\mathrm{T}}}{||E_1p_2||^2}$. Similar to building the residual matrix product $E_1^{\mathrm{T}}F_1F_1^{\mathrm{T}}E_1$, we repeat step (1) to step (4) with $\rho_1|0\rangle\langle 0|$ as the input and $\mathrm{e}^{\mathrm{i}\rho_{U_2}t}$ as the controlled unitary, then get the density matrix ρ_2 corresponding to the residual matrix product $E_2^{\mathrm{T}}F_2F_2^{\mathrm{T}}E_2$. In the ith iteration, we take $\rho_{i-1}|0\rangle\langle 0|$ as the input and $\mathrm{e}^{\mathrm{i}\rho_{U_i}t}$ as the controlled unitary and repeat step (1) to step (4) to get the density matrix ρ_i corresponding to the residual matrix product $E_i^{\mathrm{T}}F_iF_i^{\mathrm{T}}E_i$. Algorithm 2 describes the density matrix product method.

Algorithm 2 Quantum density matrix product method

Input: the density matrix ρ_{U_1} for $U_1U_1^{\mathrm{T}}$, the density matrix ρ_0 for $E_0^{\mathrm{T}}F_0F_0^{\mathrm{T}}E_0$, the number of iterations l.

Output: the residual matrix product ρ_l.

1: Iteration $i = 1$.
2: **while** $i \le l$ **do**
3: Perform phase estimation on ρ_{i-1} with $\mathrm{e}^{\mathrm{i}\rho_{U_i}t}$ as the controlled unitary and get ρ_{i-1}'.
4: Do controlled rotation operation on ρ_{i-1}' and get ρ_{i-1}''.
5: Perform inverse phase estimation and measurement on ρ_{i-1}'' and get ρ_i.
6: $i = i+1$
7: **return** the residual matrix product ρ_l.

Figure 6 shows the circuit of building the residual matrix product. The circuit includes l models, and the i-th model is used to construct the density matrix ρ_i.

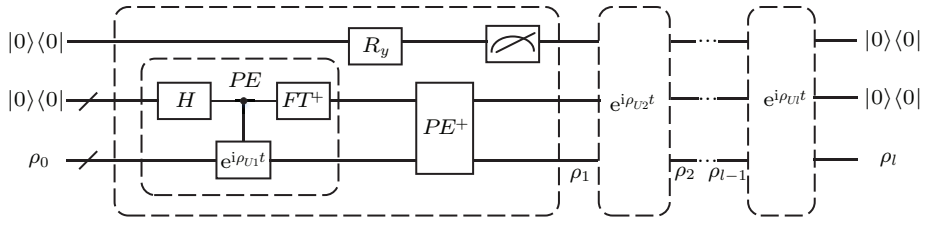


Fig. 6. Circuit of building the residual matrix product. The circuit includes l models, where the ith model is used to build the quantum state of the ith residual matrix product. Each module comprises phase estimation PE, inverse phase estimation PE^+, and the controlled rotation operation R_y, where phase estimation PE contains Hadamard operation H, the controlled unitary $\mathrm{e}^{\mathrm{i}\rho_{U_1}t}$, and inverse quantum Fourier transform FT^+. ρ_i represents the residual matrix product built in the ith iteration. Due to each module has the similar circuit structure, only the first module shows the complete circuit, and the remaining modules are represented by the label $\mathrm{e}^{\mathrm{i}\rho_{U_i}t}$ in the dotted box.

3.5. Prediction

Through l iterations, we get all the principal axes $|u_1^i\rangle$ and the regression parameters $|v_1^i\rangle$. For a new independent variable x^*, we can predict the dependent variable $\widetilde{y}^*$ according to the prediction equation $\widetilde{y}^* = \sum_{i=1}^{l} x^* p_i r_i^{\mathrm{T}}$. By defining $\widetilde{y}_i^* = x^* p_i r_i^{\mathrm{T}}$, the prediction function can be written as $\widetilde{y}^* = \sum_{i=1}^{l} \widetilde{y}_i^*$. Let $|\varphi_x^*\rangle$ denote the quantum state of the independent variable x^*. The quantum state $|\varphi_{\widetilde{y}_i}^*\rangle$ of the dependent variable $\widetilde{y}_i^*$ can be predicted based on the quantum prediction function $|\varphi_{\widetilde{y}_i}^*\rangle = \frac{1}{||h_i||^2}\langle\varphi_x^*|u_1^i\rangle|v_1^i\rangle$. The quantum prediction process is described as follows. Firstly, we add an ancilla register and perform controlled swap test operation on $|\varphi_x^*\rangle$ and $|u_1^i\rangle$. After measuring the ancilla qubit, we adopt the probability that the ancilla qubit is $|1\rangle$ to compute $\langle\varphi_x^*|u_1^i\rangle$, then get $k_i = \frac{1}{||h_i||^2}\langle\varphi_x^*|u_1^i\rangle$ by multiplying $\frac{1}{||h_i||^2}$ and $\langle\varphi_x^*|u_1^i\rangle$. Finally, we perform controlled rotation to obtain $|\varphi_{\widetilde{y}^*}\rangle = \frac{1}{\sqrt{\sum_{i=1}^{l} k_i^2}} \sum_{i=1}^{l} k_i |v_1^i\rangle$, which is the quantum state of the predicted dependent variable $\widetilde{y}_i^*$.

4. Complexity analysis

In this section, we analyze the space and time complexities of the QPLS regression algorithm, which is mainly determined by principal axes computation, building regression parameters, and residual matrices construction.

4.1. Space complexity analysis

In the first stage, we prepare the quantum state $|\phi\rangle$ as the input, which needs $O(\log(nw))$ qubits. Let ε denote the error of phase estimation. Phase estimation requires $O(\log(1/\varepsilon))$ qubits to store the estimated eigenvalues.[42] Assume amplitude estimation and phase estimation have the same error ε, amplitude estimation needs $O(\log(1/\varepsilon))$ qubits to store the estimated amplitude θ. We get the space complexity $O(\log(nw/\varepsilon^2))$ of the first stage by adding all space complexities together. For l iterations, the total space complexity is $O(\log(nw/\varepsilon^2)l)$.

In the second stage, we first adopt the same operations as the first stage to build the regression parameters r_1'; thus, the space complexity is $O(\log(nw/\varepsilon^2))$. Compared with the first stage, we need to compute the coefficient $||E_0 p_1||^2$ further. $|\varphi_{E_0}\rangle_{AB}$ and $|u_1\rangle_C$ need $O(\log(nm))$ and $O(\log(m))$ qubits, respectively, so the space complexity of calculating $||E_0 p_1||^2$ is $O(\log(nm)+\log(m))$. Since ε is very small, $O(\log(nm)+\log(m))$ can be negligible compared with $O(\log(nw/\varepsilon^2))$, and the space complexity of this stage is $O(\log(nw/\varepsilon^2))$. For l iterations, the total space complexity is $O(\log(nw/\varepsilon^2)l)$.

In the third stage, preparation the state ρ_0 and phase estimation need $O(\log(nw))$ and $O(\log(1/\varepsilon))$ qubits, respectively. Therefore, the space complexity of this stage is $O(\log(nw/\varepsilon))$. Since the space complexity of state preparation and phase estimation is the same in each iteration, the total space complexity for l iterations is $O(\log(nw/\varepsilon)l)$.

Table 1 shows the space complexities of the QPLS regression and the PLS regression. Consider that ε is usually very small, the space complexity of the third stage can be negligible. We add all space complexities together and get the total space complexity $\widetilde{O}(\log(nw/\varepsilon^2)l)$, logarithmic in the independent variable dimension n and the dependent variable dimension w.

Table 1. Space complexity.

Stage	QPLS regression	PLS regression
Stage 1	$O(\log(nw/\varepsilon^2)l)$	$O(\mathrm{poly}(nw)l)$
Stage 2	$O(\log(nw/\varepsilon^2)l)$	$O(\mathrm{poly}(nw)l)$
Stage 3	$O(\log(nw/\varepsilon)l)$	$O(\mathrm{poly}(nw)l)$

Finally, we adopt controlled swap test operation to predict the dependent variable y^* for the new independent variable x^*. Due to $|\phi_x^*\rangle$ is an n-dimensional vector, the space complexity of controlled swap test is $O(\log(n))$. We need to measure the ancilla resister $O(P(1)(1-P(1))/\varepsilon_1^2)$ times to ensure the error ε_1, where $P(1)$ is the probability that the measurement result is $|1\rangle$. Since $P(1)$ can be viewed as a constant and l controlled swap test operations are required, the space complexity of the prediction stage is $O(\log(n)l/\varepsilon_1^2)$.

4.2. Time complexity analysis

Now, we analyze the time complexity of the QPLS regression. Since the time complexity of preparing the input state $|\phi\rangle$ is $O(\mathrm{polylog}(nw))$, the total time complexity of state preparation is $O(\mathrm{polylog}(nw)l)$ for l iterations. In the first stage, the time complexity is dominated by phase estimation and the quantum binary search. Phase estimation adopts Hamiltonian evolution[8] to build the unitary operation $\mathrm{e}^{\mathrm{i}\rho_0 t} = \mathrm{e}^{\mathrm{i}\rho_0 K\Delta t}$.[26] where t is the evolution time, Δt is the time interval, and K is the number of time intervals. According to Suzuki–Trotter expansion,[23] the error of $\mathrm{e}^{\mathrm{i}\rho_0\Delta t}$ is $O(\Delta t^2)$, and the maximum error of $\mathrm{e}^{\mathrm{i}\rho_0 k\Delta t}$ $(k=0,\ldots,K-1)$ is $O(t^2/K)$. Since phase estimation needs K copies of ρ_0 and the time to access ρ_0 is $O(\mathrm{polylog}(nw))$, the time complexity of phase estimation is $O(\varepsilon^{-1}t^2\mathrm{polylog}(nw))$. The relative error of the eigenvalue λ_i satisfies $O(\lambda_i/t) \le O(\lambda_{\max}/t)$, where $\lambda_{\max}$ represents the largest eigenvalue.[43] Since the evolution time t should be $O(1/\varepsilon)$ to ensure the relative error of $O(\varepsilon)$, the time complexity of phase estimation can be further written as $O(\varepsilon^{-3}\mathrm{polylog}(nw))$. The time complexity of the quantum binary search is dominated by amplitude estimation. The time complexity of amplitude estimation is $O((2+1/2\xi)t_{G_1}/\varepsilon)$,[43] where t_{G_1} is the run time of G_1 operation, $(2+1/2\xi)$ is a constant determined by the success probability of amplitude estimation. Similar to phase estimation, t_{G_1} is $O(\varepsilon^{-3}\mathrm{polylog}(nw))$. Amplitude estimation needs

to be performed $l_s = \log(1/\varepsilon)$ times, thus the time complexity of this stage is $O(\log(1/\varepsilon)\varepsilon^{-4}\text{polylog}(nw))$. For l iterations, the total time complexity of this stage is $O(\varepsilon^{-3}\text{polylog}(nw)l + \log(1/\varepsilon)\varepsilon^{-4}\text{polylog}(nw)l)$.

In the second stage, the time complexity of computing the parameter r_1' is the same as the first stage. The time complexity of computing $||u_1||^2$ is dominated by CNOT and H operations and can be negligible. Therefore, the total time complexity of this stage for l iterations is also $O(\varepsilon^{-3}\text{polylog}(nw)l + \log(1/\varepsilon)\varepsilon^{-4}\text{polylog}(nw)l)$.

In the third stage, the time complexity is dominated by phase estimation. Due to the time complexity of preparing ρ_{U_i} is $O(\text{polylog}(nm))$ and the error is ε, the time complexity of phase estimation is $O(\varepsilon^{-3}\text{polylog}(nm))$. Furthermore, $O\left(\frac{\max_k(1-\widetilde{\lambda}_k^1)^2}{\min_k(1-\widetilde{\lambda}_k^1)^2}\right)$ measurements are performed to get the density matrix ρ_1. Thus, the time complexity of building ρ_1 is $O\left(\frac{\max_k(1-\widetilde{\lambda}_k^1)^2}{\min_k(1-\widetilde{\lambda}_k^1)^2}\varepsilon^{-3}\text{polylog}(nm)\right)$. If measurement operation is replaced by amplitude amplification, the time complexity is reduced to $O\left(\frac{\max_k(1-\widetilde{\lambda}_k^1)}{\min_k(1-\widetilde{\lambda}_k^1)}\varepsilon^{-3}\text{polylog}(nm)\right)$. Since only the first iteration can adopt amplitude amplification, the time complexity of preparing ρ_i $(i \geq 2)$ is $O\Big(\text{polylog}(nm)\cdot\varepsilon^{-3}\cdot \frac{\max_k(1-\widetilde{\lambda}_k^1)}{\min_k(1-\widetilde{\lambda}_k^1)}\cdot\prod_{j=i}^{l}\left(\frac{\max_k(1-\widetilde{\lambda}_k^j)}{\min_k(1-\widetilde{\lambda}_k^j)}\right)^2\Big)$, where $\widetilde{\lambda}_k^j$ represents the k-th eigenvalue of ρ_{U_j}. The total time complexity for l iterations is $O\Big(\text{polylog}(nm)\cdot\sum_{i=1}^{l}\varepsilon^{-3}\cdot\frac{\max_k(1-\widetilde{\lambda}_k^1)}{\min_k(1-\widetilde{\lambda}_k^1)}\cdot\prod_{j=2}^{i}\left(\frac{\max_k(1-\widetilde{\lambda}_k^j)}{\min_k(1-\widetilde{\lambda}_k^j)}\right)^2\Big)$.

Table 2 shows the time complexities of the QPLS regression and the PLS regression. For the QPLS regression, due to ε is generally small, the time complexity of state preparation can be negligible. We add up the time complexities of all stages and get the total time complexity

$$O\Bigg((\log(1/\varepsilon)\varepsilon^{-1}+1)\varepsilon^{-3}\text{polylog}(nw)l + \varepsilon^{-3}\text{polylog}(nm)\sum_{i=1}^{l}\cdot\frac{\max_k(1-\widetilde{\lambda}_k^1)}{\min_k(1-\widetilde{\lambda}_k^1)} \times\prod_{j=2}^{i}\left(\frac{\max_k(1-\widetilde{\lambda}_k^j)}{\min_k(1-\widetilde{\lambda}_k^j)}\right)^2\Bigg). \tag{22}$$

For the PLS regression, the total time complexity is $O(\text{poly}(nw)l + \text{poly}(nm)l)$. The QPLS regression achieves exponential speed-ups over the PLS regression on the independent variable dimension n, the dependent variable dimension w, and the number of variables m when $1/\varepsilon$ and l are $O(\text{poly}(nw))$ and $\frac{\max_k(1-\widetilde{\lambda}_k^1)}{\min_k(1-\widetilde{\lambda}_k^1)}$ is $O(\text{poly}(nm))$.

Table 2. Time complexity.

Stage	QPLS regression	PLS regression
Stage 1	$O(\log(1/\varepsilon)\varepsilon^{-4}\text{polylog}(nw)l+\varepsilon^{-3}\text{polylog}(nw)l)$	$O(\text{poly}(nw)l)$
Stage 2	$O(\log(1/\varepsilon)\varepsilon^{-4}\text{polylog}(nw)l+\varepsilon^{-3}\text{polylog}(nw)l)$	$O(\text{poly}(nw)l)$
Stage 3	$O\Big(\text{polylog}(nm)\cdot\sum_{i=1}^{l}\varepsilon^{-3}\cdot\frac{\max_k(1-\widetilde{\lambda}_k^1)}{\min_k(1-\widetilde{\lambda}_k^1)}\cdot\prod_{j=2}^{i}\left(\frac{\max_k(1-\widetilde{\lambda}_k^j)}{\min_k(1-\widetilde{\lambda}_k^j)}\right)^2\Big)$	$O(\text{poly}(nm)l)$

In the prediction stage, the time complexity of controlled swap test operation is $O(1)$, the inner product $\langle\varphi_x^*|u_1^i\rangle$ can reach accuracy ε_1 by iterating controlled swap test operation $O(P(1)(1-P(1))/\varepsilon_1^2)$ times, where $P(1)$ is the probability that the measurement result is $|1\rangle$. Therefore, the total time complexity of the prediction stage is $O(lP(1)(1-P(1))/\varepsilon_1^2)$. Since $P(1)$ is a constant, the time complexity can be further written as $O(l/\varepsilon_1^2)$, which is determined by the error ε_1.

5. Conclusion and future work

In this paper, we proposed a quantum PLS regression algorithm and analyzed the space and time complexities. Aiming at the problem of high computational complexity for the PLS regression algorithm, we designed the quantum eigenvector search method, which skillfully adopted the quantum comparison operations to speed up the building of the principal axis p_i. In addition, we combined the quantum eigenvector search method and the Bell-basis measurement method to construct the regression parameters r_i. Finally, we designed the density matrix product method to produce the residual matrix product $E_i^{\mathrm{T}}F_iF_i^{\mathrm{T}}E_i$, which avoided multiple accesses to QRAM and improved the execution speed of the algorithm. Furthermore, the complexity analysis shows that the QPLS regression algorithm achieves exponential speed-ups over the PLS regression on the independent variable dimension n, the dependent variable dimension w, and the number of variables m when $1/\varepsilon$ and l are $O(\text{poly}(nw))$ and $\frac{\max_k(1-\widetilde{\lambda}_k^1)}{\min_k(1-\widetilde{\lambda}_k^1)}$ is $O(\text{poly}(nm))$. The quantum eigenvector search method is a more promising method to implement the specified state search of quantum algorithms. We hope that the QPLS regression algorithm can inspire more fruitful results in the quantum machine learning field.

Acknowledgements

Project supported by the Fundamental Research Funds for the Central Universities, China (Grant No. 2019XD-A02), the National Natural Science Foundation of China (Grant Nos. U1636106, 61671087, 61170272, and 92046001), Natural Science Foundation of Beijing Municipality, China (Grant

No. 4182006), Technological Special Project of Guizhou Province, China (Grant No. 20183001), and the Foundation of Guizhou Provincial Key Laboratory of Public Big Data (Grant Nos. 2018BDKFJJ016 and 2018BDKFJJ018).

References

[1] Krzywinski M and Altman N 2015 *Nature Methods* **12** 1103
[2] Momma M and Bennett K P 2003 *Learning Theory and Kernel Machines* **2777** 216
[3] Li B M, Hu M L and Fan H 2019 *Acta Phys. Sin.* **68** 030304 (in Chinese)
[4] Kong X Y, Zhu Y Y, Wen J W, Xin T, Li K R and Long G L 2018 *Acta Phys. Sin.* **67** 220301 (in Chinese)
[5] Kerenidis I and Prakash A 2020 *Phys. Rev. A* **101** 022316
[6] Schuld M, Fingerhuth M and Petruccione F 2017 *Europhys. Lett.* **119** 60002
[7] Schuld M and Petruccione F 2018 *Sci. Rep.* **8** 2772
[8] Zhao Z K, Fitzsimons J K and Fitzsimons J F 2019 *Phys. Rev. A* **99** 052331
[9] Yu C H, Gao F and Wen Q Y 2021 *IEEE Transactions on Knowledge and Data Engineering* **33** 858
[10] Hou Y Y, Li J, Chen X B, Li H J, Li C Y, Tian Y, Li L L, Cao Z W and Wang N 2020 *Quantum Inf. Process.* **19** 278
[11] Song X T, Li H W, Yin Z Q, Liang W Y, Zhang C M, Han Y G, Chen W and Han Z F 2015 *Chin. Phys. Lett.* **32** 80302
[12] Li Q, He Y and Jiang J P 2011 *Quantum Inf. Process.* **10** 13
[13] Lloyd S, Mohseni M and Rebentrost P 2014 *Nat. Phys.* **10** 631
[14] Chen X Y 2015 *Chin. Phys. Lett.* **32** 10301
[15] Cao X and Shang Y 2014 *Chin. Phys. Lett.* **31** 110302
[16] Daskin A 2016 *Quantum Inf. Process.* **15** 4013
[17] Yu C H, Gao F, Wang Q L and Wen Q Y 2016 *Phys. Rev. A* **94** 042311
[18] Rozema L A, Mahler D H, Hayat A, Turner P S and Steinberg A M 2014 *Phys. Rev. Lett.* **113** 160504
[19] Yang Y X, Chiribella G and Hayashi M 2016 *Phys. Rev. Lett.* **117** 090502
[20] Yang Y X, Chiribella G and Ebler D 2016 *Phys. Rev. Lett.* **116** 080501
[21] Chai G, Cao Z, Liu W, Wang S, Huang P and Zeng G 2019 *Phys. Rev. A* **99** 032326
[22] Biamonte J, Wittek W, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 *Nature* **549** 195
[23] Dunjko V and Briegel H J 2018 *Reports on Progress in Physics* **81** 074001
[24] Duan B J, Yuan J B, Liu Y and Li D 2017 *Phys. Rev. A* **96** 032301
[25] Lu T C, Yu G R and Juang J C 2013 *IEEE Transactions on Neural Networks and Learning Systems* **24** 1266
[26] Carleo G and Troyer M 2017 *Science* **355** 602
[27] Rebentrost P, Bromley T R, Weedbrook C and Lloyd S 2018 *Phys. Rev. A* **98** 042308
[28] Cong I, Choi S and Lukin M D 2019 *Nat. Phys.* **15** 1273
[29] Sun J, Lu S F and Liu F 2014 *Chin. Phys. Lett.* **31** 070304
[30] Harrow A W, Hassidim A and Lloyd S 2009 *Phys. Rev. Lett.* **103** 150502
[31] Wiebe N, Braun D and Lloyd S 2012 *Phys. Rev. Letter* **109** 050505
[32] Schuld M, Sinayskiy I and Petruccione F 2016 *Phys. Rev. A* **94** 022342
[33] Wang G M 2012 *Phys. Rev. A* **96** 012335
[34] Giovannetti V, Lloyd S and Maccone L 2008 *Phys. Rev. Lett.* **100** 160501
[35] Schuld M and Petruccione F 2018 *Sci. Rep.* **8** 2772
[36] Lloyd S, Mohseni M and Rebentrost P 2014 *Nat. Phys.* **10** 631
[37] Yu C H, Gao F, Lin S and Wang J B 2019 *Quantum Inf. Process.* **18** 249
[38] Yang L, Li K and Dai H Y 2019 *Acta Phys. Sin.* **68** 140301 (in Chinese)
[39] Chen X, Zhang Z W and Zhao H 2016 *Chin. Phys. Lett.* **33** 104203
[40] Cincio L, Subasi Y, Sornborger A T and Coles P J 2018 *New J. Phys.* **20** 113022
[41] Sijmen D J 1993 *Chemometrics and Intelligent Laboratory Systems* **18** 251
[42] Duan B J, Yuan J B, Xu J and Li D 2018 *Phys. Rev. A* **99** 032311
[43] Brassard G, Hoyer P, Mosca M and Tapp A 2002 *Contemporary Mathematics* **305** 53