

Quantum algorithm and quantum circuit for A-Optimal Projection: dimensionality reduction

Bojia Duan* and Jiabin Yuan†

*College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics,
No.29 Jiangjun Avenue, 211106 Nanjing, China.*

(Dated: October 30, 2019)

Abstract

Learning low dimensional representation is a crucial issue for many machine learning tasks such as pattern recognition and image retrieval. In this article, we present a quantum algorithm and a quantum circuit to efficiently perform A-Optimal Projection for dimensionality reduction. Compared with the best-know classical algorithms, the quantum algorithm shows an exponential speedup in both the number of training vectors n and the reduced feature space dimension k . We show that the space and time complexity of the QAOP circuit are $O[\log_2(nk/\epsilon)]$ and $O[\log_2(nk)poly(\log_2\epsilon^{-1})]$ respectively, with fidelity at least $1 - \epsilon$. Firstly, a reformation of the original QAOP algorithm is proposed to help omit the quantum-classical interactions during the QAOP algorithm. Then the quantum algorithm and quantum circuit with performance guarantees are proposed. Specifically, the quantum circuit modules for preparing the initial quantum state and implementing the controlled rotation can be also used for other quantum machine learning algorithms.

* deja@nuaa.edu.cn

† jbyuan@nuaa.edu.cn

I. INTRODUCTION

Learning low dimensional image representations has gained significant importance in many image processing tasks such as recognition and retrieval [1–3]. A range of applications of this problem can be seen in the field of medical imaging such as liver cirrhosis, lung cancer classification, breast cancer diagnosis [3]. Another typical example is face recognition which is typically used in security systems or as a commercial identification and marketing tool [4]. Recent studies have shown that images are possibly sampled from a low dimensional manifold, however, the visual features, such as color, texture, shape, which are usually extracted for the image representation, are usually of very high dimensionality [5]. Therefore, a range of techniques have been developed for dimensionality reduction. For instance, principal component analysis (PCA) is guaranteed in terms of the linearly embedded manifold [6]. Moreover, Isomap, Locally Linear Embedding, and Laplacian Eigenmap are proposed for nonlinear embedded manifold [7–9].

Different from all the aforementioned techniques which are not directly related to the regression task, X. He proposed a novel dimensionality reduction algorithm named A-Optimal Projection (AOP) which performs better regression performance in the reduced space [5]. This approach can be performed under either unsupervised or supervised mode, which is a more widely used algorithm compared to the unsupervised algorithm PCA or the supervised algorithm linear discriminant analysis (LDA) [6]. Moreover, different from most dimensionality reduction algorithms which are applied as pre-processing of the data, AOP can directly improve the performance of a regression model in the reduced space, therefore, the learned regression model can be as stable as possible.

Time complexity is a significant drawback in classical dimensionality reduction algorithms. A range of quantum algorithms have achieved exponentially speed up of classical quantum machine learning algorithms compared with the classical ones [10–15]. In the application field of quantum dimensionality reduction, quantum algorithm for PCA are proposed for unsupervised mode [11], and quantum algorithm for LDA are proposed for supervised mode and classification [12]. In this paper, we propose a quantum algorithm for AOP, which achieves exponentially speed up compared with the classical polylogarithmic in both m , the number of training vectors, and k , the reduced feature space dimension.

Our work has two major contributions. First, we present a quantum algorithm for solving

the learning process of the AOP algorithm. A reformulation of the original classical AOP algorithm is introduced here which helps the QAOP algorithm be implemented more efficiently. The quantum algorithm are made of iterations, where each iteration mainly consists of phase estimation and a controlled rotation. The reformulated AOP and the partial trace technology can help omit the quantum-classical interactions during the quantum algorithm. Second, we design the detailed quantum circuit for the proposed QAOP algorithm which makes it possible to execute the QAOP algorithm on a universal quantum computer. The circuit for preparing the initial state are presented and the detailed circuit for the controlled rotation are designed. The space and time analysis of the quantum circuit also shows an exponential speedup in the size of the training set and the feature space than the classical counterparts.

This paper is arranged as follows: We give a brief overview of the classical AOP algorithm in Sec. II. In Sec. III, the quantum algorithm for AOP algorithm which is used in dimension reduction is presented. In Sec. IV, the overview and detailed quantum circuits for solving QAOP algorithm are designed. Finally we show the conclusions in Sec. V.

II. REVIEW OF CLASSICAL A-OPTIMAL PROJECTION

In this section, we briefly review the AOP model and learning algorithm.

The classical AOP dimensionality reduction aims to improve the regression performance in the reduced space which preserves similarities between the data pairs. The AOP dimensionality reduction algorithm returns the directions of projections, and with this result, the data can be projected onto a lower-dimensional subspace which can be directly used for regression problem.

Let $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_m)$ be a $n \times m$ data matrix, where m is the number of data points and n is the number of features. In the graph based dimensionality reduction, given a nearest neighbor graph G which represents the geometrical structure of the data manifold. Each vertex of the graph represents a data point. Let $\mathbf{S} \in \mathbb{R}^{m \times m}$ be the weight matrix of the graph and $N_k(\mathbf{x})$ denote the k nearest neighbors of \mathbf{x} . Then a simple example of \mathbf{S} can be defined as follows:

$$S_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in N_k(\mathbf{x}_j) \quad \text{or} \quad \mathbf{x}_j \in N_k(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

AOP aims to find a projection matrix $\mathbf{A} \in \mathbb{R}^{n \times k}$ that maps the the points \mathbf{x}_i to $\mathbf{y}_i \in \mathbb{R}^k$ ($i = 1, \dots, m$, and $k \ll n$), where $\mathbf{y}_i = \mathbf{A}^T \mathbf{x}_i$. And using \mathbf{y}_i to train a linear regression model:

$$z = \beta^T \mathbf{y} + \epsilon_0 \quad (2)$$

where z is the observation, β is the weight vector and ϵ_0 is an unknown error with Gaussian distribution.

Formally, the objective function of AOP is:

$$\min_{\mathbf{A}} Tr \left((\mathbf{A}^T \mathbf{X} (\mathbf{I} + \lambda_1 \mathbf{L}) \mathbf{X}^T \mathbf{A} + \lambda_2 \mathbf{I})^{-1} \right) \quad (3)$$

where λ_1 and λ_2 are the regularization coefficients which are very small, and $\mathbf{L} = diag(\mathbf{S}\mathbf{1}) - \mathbf{S}$ is the graph Laplacian ($\mathbf{1}$ is a vector of all ones).

To solve the objective function, Ref. [5] introduce a variables \mathbf{B} and the optimization problem (3) is equivalent to the following:

$$\min_{\mathbf{A}, \mathbf{B}} \left\| \mathbf{I} - \mathbf{A}^T \tilde{\mathbf{X}} \mathbf{B} \right\|^2 + \lambda_2 \|\mathbf{B}\|^2 \quad (4)$$

where $\tilde{\mathbf{X}} = \mathbf{X}\Sigma$, and Σ is from the cholsky decomposition: $\mathbf{I} + \lambda_1 \mathbf{L} = \Sigma \Sigma^T$.

It tells us that the optimal \mathbf{A} can be obtained by iteratively computing \mathbf{A} and \mathbf{B} . Then the overall procedure of the AOP learning algorithm is depicted as follows:

- 1) Initialize the matrix \mathbf{A} by computing the PCA of the data \mathbf{X} .
- 2) Compute the matrix \mathbf{B} according to the Eq. (5):

$$\frac{\partial \phi}{\partial \mathbf{B}^T} = 0 \Rightarrow \mathbf{B} = \left(\tilde{\mathbf{X}}^T \mathbf{A} \mathbf{A}^T \tilde{\mathbf{X}} + \lambda_2 \mathbf{I} \right)^{-1} \tilde{\mathbf{X}}^T \mathbf{A} \quad (5)$$

- 3) Update the matrix \mathbf{A} according to the Eq. (6), and normalize \mathbf{A} such that $\|\mathbf{A}\|_F \leq \rho$.

$$\frac{\partial \phi}{\partial \mathbf{A}} = 0 \Rightarrow \mathbf{A} = \left(\tilde{\mathbf{X}} \mathbf{B} \mathbf{B}^T \tilde{\mathbf{X}}^T \right)^{-1} \tilde{\mathbf{X}} \mathbf{B} \quad (6)$$

- 4) Repeat steps 2 and 3 until convergence.

III. QUANTUM A-OPTIMAL PROJECTION

In this section, we propose the quantum AOP algorithm for dimensionality reduction. We firstly reformulated the original classical AOP algorithm. And with the help of the reformulation, the proposed quantum AOP algorithm can then be implemented more efficiently.

A. Reformulation of the AOP algorithm

We reformulated the algorithm in Sec. II in terms of quantum mechanics. Firstly, we adjust the initialization of \mathbf{A} to make it closer to the optimal solution than the original algorithm. Secondly, we combine the steps 2 and 3 into one step and remove the variable \mathbf{B} . The advantage of eliminating \mathbf{B} is to help avoid quantum-classical transformation during the iteration of the algorithm. Specifically, in one iteration of the original algorithm, a quantum state is needed to be computed and sampled as to reconstruct the matrix \mathbf{B} , and then used to update \mathbf{A} . In our methods, the elimination of the matrix \mathbf{B} can help update the quantum state representing \mathbf{A} without sampling and reconstruction in one of the iterations. Finally, by introducing the partial trace technology, quantum-classical interaction can also be omitted between the iterations.

(1) Initialization of \mathbf{A} . The original AOP algorithm compute PCA of the data \mathbf{X} to initialize the matrix \mathbf{A} . In contrast, we compute PCA of $\tilde{\mathbf{X}}$ to obtain the initialization $\mathbf{A}^{(0)}$. As shown in Eq. (3), when λ_1 and λ_2 are set to be zero, the objection function (3) is equivalent to the objection function of PCA referring to the data \mathbf{X} . And when λ_2 is zero, the objection function (3) is equivalent to the objection function of PCA referring to the data $\tilde{\mathbf{X}}$. It is obvious that the later one is closer to the optimal solution than the former one.

(2) Reformulation of the AOP algorithm. Now turning to the steps 2 and 3 of the AOP algorithm in one of the iterations. Suppose the singular value decomposition of the matrix $\tilde{\mathbf{X}}$ is $\tilde{\mathbf{X}} = \sum_{j=1}^r \sigma_j |\mathbf{u}_j\rangle \langle \mathbf{v}_j|$, where $r \leq \min(m, n)$ is the rank of $\tilde{\mathbf{X}}$, and $\sigma_k (\sigma_1 > \dots > \sigma_r > 0)$ are the singular values of $\tilde{\mathbf{X}}$, with \mathbf{u}_j and \mathbf{v}_j being the left and right singular vectors. Obviously, we have $\tilde{\mathbf{X}}^T = \sum_{j=1}^r \sigma_j |\mathbf{v}_j\rangle \langle \mathbf{u}_j|$ and $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T = \sum_{j=1}^r \sigma_j^2 |\mathbf{u}_j\rangle \langle \mathbf{u}_j|$.

As $\mathbf{A}^{(0)}$ is the PCA of $\tilde{\mathbf{X}}$, we have

$$\mathbf{A}^{(0)} = \text{pca}(\tilde{\mathbf{X}}) = \sum_{j=1}^k |\mathbf{u}_j\rangle \langle \mathbf{j}|, \quad (7)$$

where k is the rank of \mathbf{A} , and $|\mathbf{j}\rangle$'s are the basis states. Now we have the theorem 1 (and the proof is shown in Appendix A.).

Theorem 1: Given the matrix $\tilde{\mathbf{X}} = \sum_{j=1}^r \sigma_j |\mathbf{u}_j\rangle \langle \mathbf{v}_j|$, the i -th iteration of the AOP

algorithm outputs the matrix $\mathbf{A}^{(i)}$:

$$\mathbf{A}^{(i)} = \sum_{j=1}^k \beta_j^{(i)} |\mathbf{u}_j\rangle \langle \mathbf{j}| = \sum_{j=1}^k \frac{(\sigma_j \beta_j^{(i-1)})^2 + \lambda_2}{\sigma_j^2 \beta_j^{(i-1)}} |\mathbf{u}_j\rangle \langle \mathbf{j}| \quad (8)$$

where $1 \leq i \leq s$ and $\beta_j^{(0)} = 1$ for all j 's.

According to the theorem 1, the reformulated AOP algorithm is presented as follows:

- 1) Initialize the matrix $\mathbf{A}^{(0)}$ by computing the PCA of $\tilde{\mathbf{X}}$ according to Eq. (7).
- 2) Update the matrix \mathbf{A} according to Eq. (8).
- 3) Repeat step 2 until convergence.

The modeling of the AOP algorithm can be shown in Fig. 1.

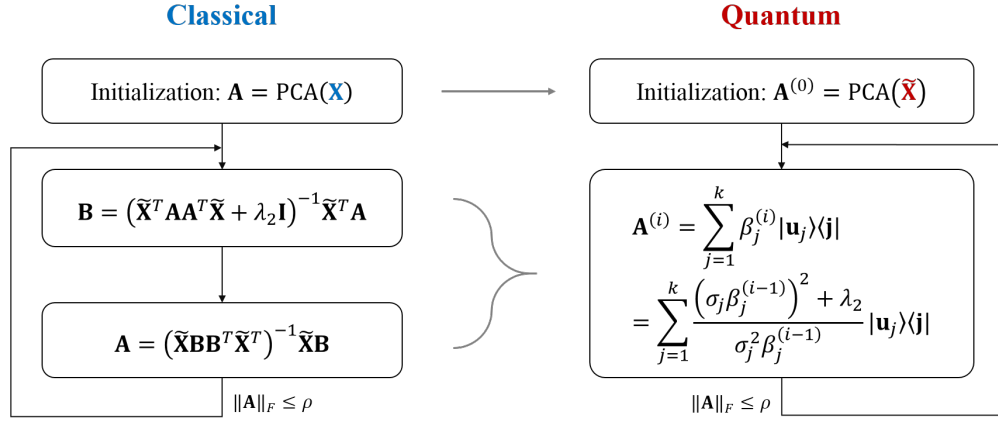


FIG. 1: Reformulation of the AOP algorithm.

B. QAOP algorithm

The overall procedure of our QAOP algorithm is then proposed as follows.

Algorithm. $\mathbf{A}^{(s)} = \text{QAOP}(\tilde{\mathbf{X}}, \lambda_2)$.

1. Initialize $i = 1$, apply quantum PCA algorithm of $\tilde{\mathbf{X}}$ to compute $\mathbf{A}^{(0)}$ [11].
2. Perform one iteration Q of the QAOP algorithm to compute $\mathbf{A}^{(i)}$, i.e. $\mathbf{A}^{(i)} = Q(\tilde{\mathbf{X}}, \mathbf{A}^{(i-1)})$.
3. Set $i = i + 1$, and repeat Steps 2 until the number of iterations $i = s$; now the projection matrix $\mathbf{A}^{(s)}$ can be obtained.

The quantum algorithm for i -th iteration Q of the QAOP algorithm in Step 2 is presented as follows.

(1) Prepare four quantum registers in the state

$$|\psi_0\rangle = |0\rangle^a (|0\rangle|0\rangle \cdots |0\rangle)^C (|0\rangle|0\rangle \cdots |0\rangle)^B (|\psi_{\mathbf{A}^{(i-1)}}\rangle)^A. \quad (9)$$

where the superscript a represents the ancilla qubit, the superscript C, B, A represents the register C, B and A , respectively.

(2) Perform the unitary operation $\mathbf{U}_{PE}(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\dagger)$ and $\mathbf{U}_{PE}(\mathbf{A}^{(i-1)}\mathbf{A}^{(i-1)\dagger})$ on the state, then we have the state

$$|\psi_1\rangle = \frac{1}{\sqrt{N_1}} |0\rangle^a \sum_{j=1}^k \beta_j^{(i-1)} |\sigma_j^2\rangle^C \left| \left(\beta_j^{(i-1)} \right)^2 \right\rangle^B |\mathbf{u}_j\rangle |\mathbf{v}_j\rangle^A. \quad (10)$$

where \mathbf{U}_{PE} representing the unitary matrix for phase estimation which we fully characterized in [15].

(3) Apply a controlled rotation \mathbf{R}_f to the ancilla qubit, controlled by both the register C and B . \mathbf{R}_f is defined as follows:

$$\begin{aligned} R_f : |0\rangle^a |\sigma_j^2\rangle^C \left| \left(\beta_j^{(i-1)} \right)^2 \right\rangle^B \\ \rightarrow \left[\rho \left(1 + \frac{\lambda_2}{\left(\sigma_j \beta_j^{(i-1)} \right)^2} \right) |1\rangle + \sqrt{1 - \rho^2 \left(1 + \frac{\lambda_2}{\left(\sigma_j \beta_j^{(i-1)} \right)^2} \right)^2} |0\rangle \right]^a |\sigma_j^2\rangle^C \left| \left(\beta_j^{(i-1)} \right)^2 \right\rangle^B. \end{aligned} \quad (11)$$

This rotation transforms the state to

$$\begin{aligned} |\psi_2\rangle = \frac{1}{\sqrt{N_1}} \left[\rho \left(1 + \frac{\lambda_2}{\left(\sigma_j \beta_j^{(i-1)} \right)^2} \right) |1\rangle + \sqrt{1 - \rho^2 \left(1 + \frac{\lambda_2}{\left(\sigma_j \beta_j^{(i-1)} \right)^2} \right)^2} |0\rangle \right]^a \\ \otimes \sum_{j=1}^k \beta_j^{(i-1)} |\sigma_j^2\rangle^C \left| \left(\beta_j^{(i-1)} \right)^2 \right\rangle^B |\mathbf{u}_j\rangle |\mathbf{v}_j\rangle^A. \end{aligned} \quad (12)$$

(4) Uncompute the registers C, B and A , remove the register C and B , and measure the ancilla qubit to be $|1\rangle$. Then, we have the state proportional to

$$\left| \psi_{\mathbf{A}}^{(i)} \right\rangle = \frac{1}{\sqrt{N_2}} \sum_{j=1}^k \frac{\left(\sigma_j \beta_j^{(i-1)} \right)^2 + \lambda_2}{\sigma_j^2 \beta_j^{(i-1)}} |\mathbf{u}_j\rangle |\mathbf{v}_j\rangle^A. \quad (13)$$

Here, we can construct the matrix $\mathbf{A}^{(i)}\mathbf{A}^{(i)\dagger}$ for the phase estimation in the next iteration by taking a partial trace of $\left| \psi_{\mathbf{A}}^{(i)} \right\rangle \left\langle \psi_{\mathbf{A}}^{(i)} \right|$. Note that the eigenvectors of $\mathbf{A}^{(i)}\mathbf{A}^{(i)\dagger}$ are \mathbf{u}_j and

the corresponding eigenvalues are $\left(\beta_j^{(i)}\right)^2$. Then the density matrix that represents $\mathbf{A}^{(i)}\mathbf{A}^{(i)\dagger}$ can be obtained [16] :

$$tr_2 \left(\left| \psi_{\mathbf{A}}^{(i)} \right\rangle \left\langle \psi_{\mathbf{A}}^{(i)} \right| \right) = \frac{1}{\sum_{k=1}^r \left(\beta_j^{(i)} \right)^2} \sum_{k=1}^r \left(\beta_j^{(i)} \right)^2 |\mathbf{u}_k\rangle \langle \mathbf{u}_k| = \frac{\mathbf{A}^{(i)}\mathbf{A}^{(i)\dagger}}{tr \left(\mathbf{A}^{(i)}\mathbf{A}^{(i)\dagger} \right)}. \quad (14)$$

IV. QAOP CIRCUIT

In this section, we study the QAOP algorithm in terms of the quantum circuit model. The quantum circuits provide the possibility to implement the quantum algorithm on a universal quantum computer. First, we presents the overview model of the QAOP circuit. Second, we study in depth the realization of the initial state preparation and the controlled rotation in terms of the quantum circuit model. Finally, the space and time resources required for the quantum circuit are analyzed.

The overview of the circuit for solving QAOP is shown in Fig. 2. It provides one model to implement the QAOP algorithm. Take the i -th iteration of the QAOP algorithm for example. It can be divided into three major steps: (1) Phase estimation: as the eigenspace of the unitary $e^{-i\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T t_0}$ and $e^{-i\mathbf{A}^{(i-1)}(\mathbf{A}^{(i-1)})^T t_0}$ are both spanned by the eigenvectors $|\mathbf{u}_j\rangle$, they can be both implemented on the input state $|\psi_{\mathbf{A}}^{(i-1)}\rangle$. And the Hadamard gates and the inverse QFT of $\mathbf{U}_{PE}(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\dagger)$ and $\mathbf{U}_{PE}(\mathbf{A}^{(i-1)}\mathbf{A}^{(i-1)\dagger})$ can be implemented in parallel. (2) Controlled rotation: it consists of $U_{\beta,\sigma}$ and $c - R_y$. Firstly the operation $U_{\beta,\sigma}$ computes the function y_k of the output eigenvalues of $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\dagger$ and $\mathbf{A}^{(i-1)}\mathbf{A}^{(i-1)\dagger}$ as follows:

$$y_j = 1 + \frac{\lambda_2}{\left(\sigma_j \beta_j^{(i-1)}\right)^2}. \quad (15)$$

And then the operation $c - R_y$ extracts the value of y_j in the basic states of the register L to the amplitude of the ancilla qubit. (3) Uncomputing: undo the Reg. B, C and L , and measure the top ancilla qubit. If the result returns to 1, then the Reg. A of the quantum system collapses to the output state $|\psi_{\mathbf{A}}^{(i)}\rangle$, which is also the input state of the $(i+1)$ -th iteration.

We now deal with the detailed QAOP circuit. In the following, we mainly investigate the quantum circuits for the initial state preparation and the controlled rotation in one iteration

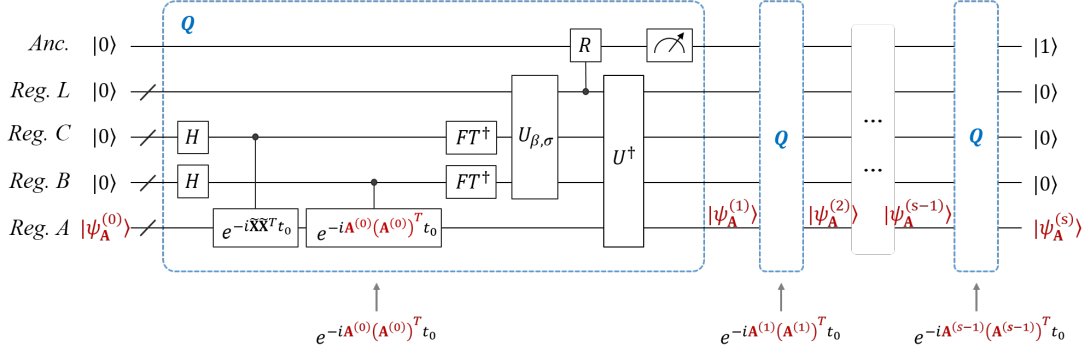


FIG. 2: Overview of the quantum circuit for solving the reformulated AOP. Wires with '/' represent the groups of qubits. The label Q in the dotted box represents one iteration of QAOP algorithm.

of the QAOP circuit.

A. State preparation

At the very beginning, we present a detailed quantum circuit for preparing the initial state of the QAOP algorithm. Suppose each elements of $\mathbf{A}^{(0)} \in \mathbb{R}^{n \times k}$ are given, and its corresponding quantum state $|\psi_{\mathbf{A}^{(0)}}\rangle = |a_1 a_2 \cdots a_q\rangle$ is a q -qubit quantum state, where $q = O[\log_2(nk)]$. Refers to [17], the quantum circuit for the initial state preparation can be shown in Fig. 3. Here, we introduce QRAM to omit the register $|\bar{\psi}\rangle$ in [17].

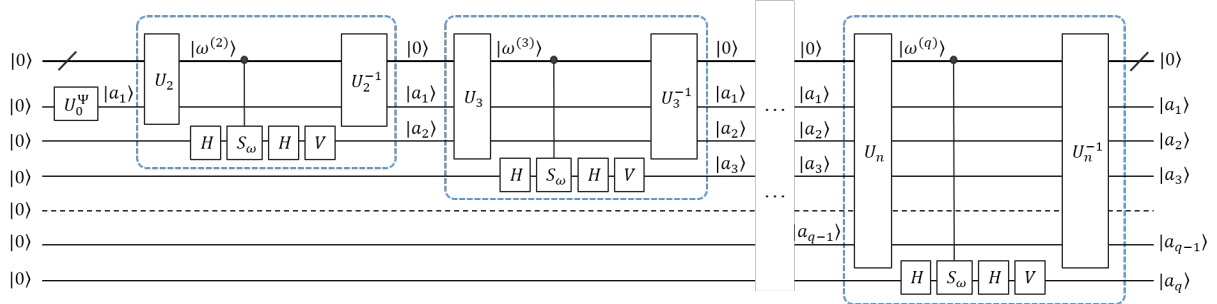


FIG. 3: The circuit for state preparation.

In Fig. 3, an register of $p = O(\log_2 \epsilon^{-1})$ qubits are used for storing the $\omega^{(i)}$, where $\omega^{(i)}$ is

short for $\omega_{a_1 \dots a_{i-1}}^{(i)}$ ($i = 1, 2, 3, \dots, q$) satisfying:

$$\cos^2 \left(2\pi \omega_{a_1 \dots a_{i-1}}^{(i)} \right) = \left(\frac{\alpha_{a_1 a_2 \dots a_{i-1} 0}}{\alpha_{a_1 a_2 \dots a_{i-1} 1}} \right)^2 + O(\text{poly}(\epsilon)). \quad (16)$$

All the $\omega_{a_1 \dots a_{i-1}}^{(i)}$ can be computed classically and they are supposed to be stored in the QRAM. Given the index $a_1 a_2 \dots a_{i-1}$, define $j = h(a_1 a_2 \dots a_{i-1})$ being the address where the data $\omega_{a_1 a_2 \dots a_{i-1}}^{(i)}$ stores, where $h(\cdot)$ is a hash function mapping $a_1 a_2 \dots a_{i-1}$ to j . Then define the unitary operation U_i which implements the QRAM readout operation:

$$\begin{aligned} U_i : \quad & \sum_{a_1, a_2, \dots, a_{i-1} \in \{0,1\}} \alpha_{a_1 a_2 \dots a_{i-1}} |a_1 a_2 \dots a_{i-1}\rangle |j\rangle |0\rangle \\ \xrightarrow{\text{QRAM}} \quad & \sum_{a_1, a_2, \dots, a_{i-1} \in \{0,1\}} \alpha_{a_1 a_2 \dots a_{i-1}} |a_1 a_2 \dots a_{i-1}\rangle |j\rangle \left| \omega_{a_1 a_2 \dots a_{i-1}}^{(i)} \right\rangle. \end{aligned} \quad (17)$$

specifically, U_i outputs the content $\omega_{a_1 a_2 \dots a_{i-1}}^{(i)}$ of the j -th memory cell in QRAM. Ref. [18] shows that this procedure can be implemented in time $O(p)$.

Moreover, we make further study on $c - S_{\omega^{(i)}}$ which is defined in Ref. [17]:

$$c - S_{\omega^{(i)}} : |\omega^{(i)}\rangle |0\rangle \rightarrow |\omega^{(i)}\rangle e^{2\pi i \omega^{(i)}} |0\rangle, |\omega^{(i)}\rangle |1\rangle \rightarrow |\omega^{(i)}\rangle e^{-2\pi i \omega^{(i)}} |1\rangle. \quad (18)$$

Define $R_l = \begin{pmatrix} e^{2\pi i / 2^l} & 0 \\ 0 & e^{-2\pi i / 2^l} \end{pmatrix}$, then the controlled unitary $c - R_l$ implements the following transformation:

$$\begin{aligned} |\omega_l^{(i)}\rangle |0\rangle & \rightarrow e^{2\pi i \omega_l^{(i)} / 2^l} |\omega_l^{(i)}\rangle |0\rangle \\ |\omega_l^{(i)}\rangle |1\rangle & \rightarrow e^{-2\pi i \omega_l^{(i)} / 2^l} |\omega_l^{(i)}\rangle |1\rangle \end{aligned} \quad (19)$$

where $\omega_l^{(i)}$ is the l -th binary bit of $\omega^{(i)}$, specifically, $\omega^{(i)} = 2^{-1}\omega_1^{(i)} + 2^{-2}\omega_2^{(i)} + \dots + 2^{-p}\omega_p^{(i)} = 0.\omega_1^{(i)}\omega_2^{(i)}\dots\omega_p^{(i)}$. Now $\prod_{l=1}^m (c - R_l)$ achieves the function of $c - S_{\omega^{(i)}}$:

$$\begin{aligned} |\omega_1^{(i)}\rangle |\omega_2^{(i)}\rangle \dots |\omega_p^{(i)}\rangle |0\rangle & \rightarrow e^{i2\pi(0.\omega_1^{(i)}\omega_2^{(i)}\dots\omega_p^{(i)})} |\omega_1^{(i)}\rangle |\omega_2^{(i)}\rangle \dots |\omega_p^{(i)}\rangle |0\rangle \\ |\omega_1^{(i)}\rangle |\omega_2^{(i)}\rangle \dots |\omega_p^{(i)}\rangle |1\rangle & \rightarrow e^{-i2\pi(0.\omega_1^{(i)}\omega_2^{(i)}\dots\omega_p^{(i)})} |\omega_1^{(i)}\rangle |\omega_2^{(i)}\rangle \dots |\omega_p^{(i)}\rangle |1\rangle \end{aligned} \quad (20)$$

Therefore, the quantum circuit for $c - S_{\omega^{(i)}}$ can be implemented as shown in Fig. 4.

Now we can simply infer that the number of qubits needed for preparing the initial quantum state $|\psi_{\mathbf{A}(0)}\rangle$ is $O(p + q)$, and the gates required is $O(pq)$.

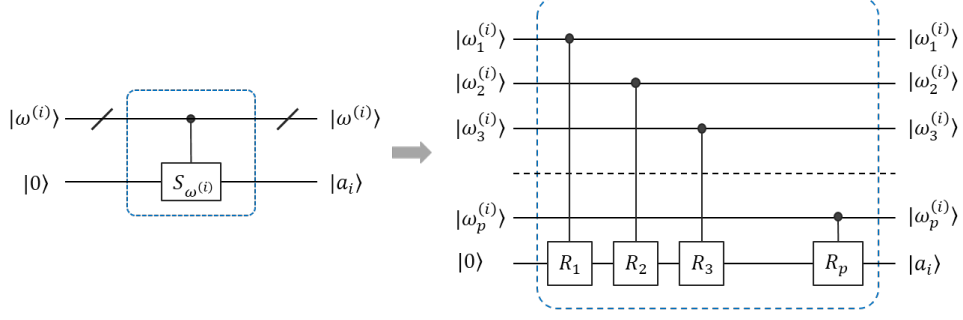


FIG. 4: The circuit for $c - S_{\omega^{(i)}}$, where $|a_i\rangle = \frac{\alpha_{a_1 \dots a_{i-1} 0}}{\alpha_{a_1 \dots a_{i-1}}} |0\rangle + \frac{\alpha_{a_1 \dots a_{i-1} 1}}{\alpha_{a_1 \dots a_{i-1}}} |1\rangle$.

In summary, with $V = \begin{pmatrix} 1 & 0 \\ 0 & -\iota \end{pmatrix}$, the unitary $(I \otimes V)(I \otimes H)(c - S_{\omega})(I \otimes H)$ implements the transformation:

$$\begin{aligned}
& |\omega^{(i)}\rangle |0\rangle \\
& \xrightarrow{I \otimes H} |\omega^{(i)}\rangle \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\
& \xrightarrow{c - S_{\omega}} |\omega^{(i)}\rangle \frac{e^{2\pi i \omega^{(i)}} |0\rangle + e^{-2\pi i \omega^{(i)}} |1\rangle}{\sqrt{2}} \\
& \xrightarrow{I \otimes H} |\omega^{(i)}\rangle [\cos(2\pi \omega^{(i)}) |0\rangle + \iota \sin(2\pi \omega^{(i)}) |1\rangle] \\
& \xrightarrow{I \otimes V} |\omega^{(i)}\rangle [\cos(2\pi \omega^{(i)}) |0\rangle + \sin(2\pi \omega^{(i)}) |1\rangle] \\
& = |\omega^{(i)}\rangle |a_i\rangle
\end{aligned} \tag{21}$$

B. Phase estimation

The quantum circuits for phase estimation has been introduced in Ref. [19]. We now only simply analyze the space and time resources used here. Let the efficient condition number of $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\dagger$ be κ . As $\mathbf{A}^{(0)}$ is induced by PCA of $\tilde{\mathbf{X}}$, the condition number of $\mathbf{A}^{(0)}\mathbf{A}^{(0)\dagger}$ is not greater than κ , and so as the $\mathbf{A}^{(i)}\mathbf{A}^{(i)\dagger}$ for $i = 1, \dots, s-1$. Therefore, $\sigma_j^2 \in [1/\kappa, 1]$ and $\beta_j^2 \in [1/\kappa, 1]$, and the number of qubits to store σ_j^2 in Reg. B and β_j^2 in Reg. C are both $b = O(\log_2 \kappa)$. We can also learn from [19] that $O(b^2)$ operations and two calls to the controlled-unitary black boxes are needed in the stage of phase estimation.

C. The controlled rotation

The controlled rotation mainly involves the computation of $U_{\beta,\sigma}$ and $c - R_y$. In the stage of $U_{\beta,\sigma}$, Newton's method is introduced for computing $y_j = y(\sigma_j^2, \beta_j^2) = \frac{\rho(\sigma_j^2 \beta_j^2 + \lambda_2)}{\sigma_j^2 \beta_j^2} = \rho + \frac{\rho \lambda_2}{\sigma_j^2 \beta_j^2}$, where $\rho < 1$ is used for normalization of the quantum state. The value of y_j are computed out and stored in the basis state of the register L , and the number of qubits for storing y_j is $d = O(\log_2 \kappa)$. In the stage of $c - R_y$, y_j are used as controlled qubit controlling the top ancilla qubit in Fig. 2.

For $U_{\beta,\sigma}$, let $z_j = z(\sigma_j^2 \beta_j^2) = 1/(\sigma_j^2 \beta_j^2)$, then we have $y_j = \rho + \rho \lambda_2 z_j^{(s')}$. Apply the Newton method to $f(z_j) = 1/z_j - \sigma_j^2 \beta_j^2$, we can get the Newton iteration function: $z_j^{(i+1)} = g(z_j^{(i)}) = z_j^{(i)} - \frac{f(z_j^{(i)})}{f'(z_j^{(i)})} = -\sigma_j^2 \beta_j^2 (z_j^{(i)})^2 + 2z_j^{(i)}$. Then the detailed circuit for one iteration of Newton's method is presented in Fig. 5. The number of qubits needed in the ancilla registers is $O(d)$. The number of fundamental quantum operations for implementing addition and multiplication are both $O[\text{poly}(d)]$, where the degree of the polynomial is no more than 3 [20, 21], and the number of gates for implementing shift is $O(d)$.

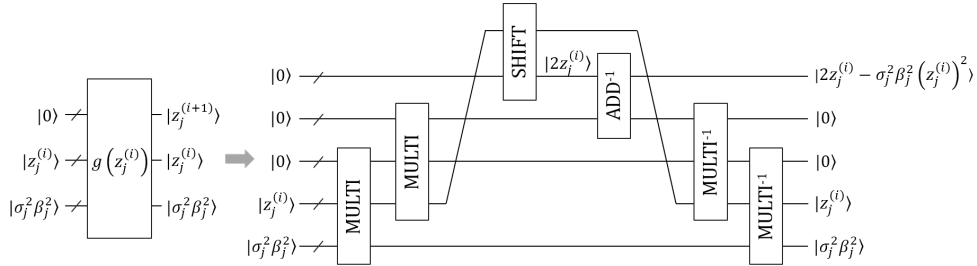


FIG. 5: The circuit for $z_j^{(i+1)} = -\sigma_j^2 \beta_j^2 (z_j^{(i)})^2 + 2z_j^{(i)}$.

The quantum circuit for $y_j = \rho + \rho \lambda_2 z_j^{(s')}$ can be simply realized as shown in Fig. 6. Both the quantum circuits can be simply realized with the quantum circuits for addition and multiplication. Therefore, the number of qubits and gates needed in the circuit are $O(d)$ and $O[\text{poly}(d)]$ respectively.

To sum up, the overall quantum circuit for $U_{\beta,\sigma}$ can be designed as shown in Fig. 7. And we can simply infer that the number of qubits needed in these circuits is $O(d + b)$. Let the number of Newton iteration be s' , then the number of gates required in Fig. 7 is $O[s' \text{poly}(d)]$.

For $c - R_y$, in order to make the output quantum state accurate, the rotation angle θ of

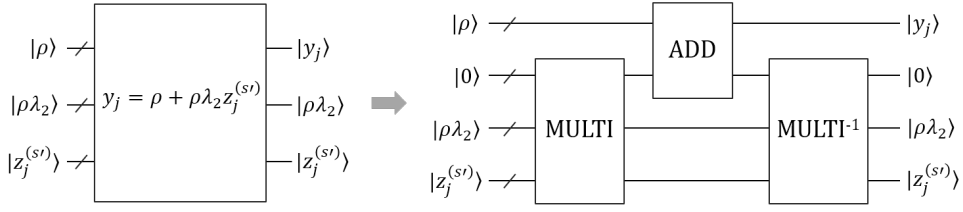


FIG. 6: The circuit for y_j .

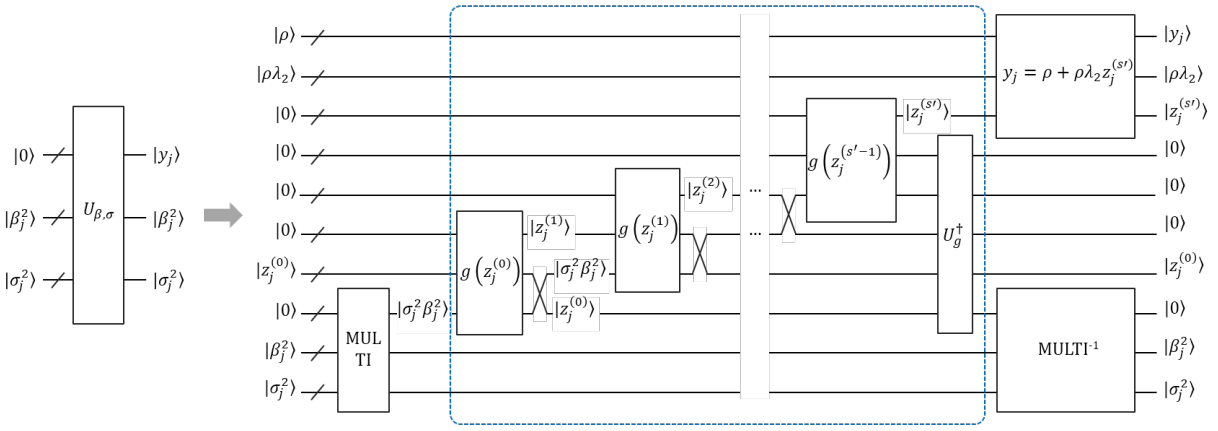


FIG. 7: The circuit for the unitary $U_{\beta,\sigma}$.

R_y satisfies $\theta = \arcsin(y)$. Since \arcsin has a convergent Taylor series, we can approximate

$$\theta = \arcsin(y) \approx y + \frac{1}{6}y^3 + \frac{3}{40}y^5 + \frac{5}{112}y^7 + \dots \quad (22)$$

Then the quantum circuit for computing the rotation angle of $c - R_y$ can be implemented as shown in Fig. 8. This circuit also only consists of the operations for addition and multiplication, therefore the number of qubits and gates required are $O(d)$ and $O[\text{poly}(d)]$ respectively.

The quantum circuit for $c - R_y$ is shown in Fig. 9, where $\theta_1, \dots, \theta_d$ are the binary bits of the output θ in Fig. 8. Obviously, the space and time complexity are both $O(d)$.

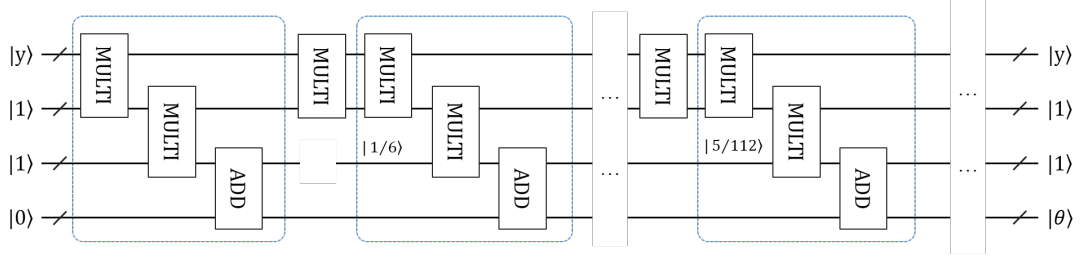


FIG. 8: The circuit for computing the rotation angle of $c - R_y$.

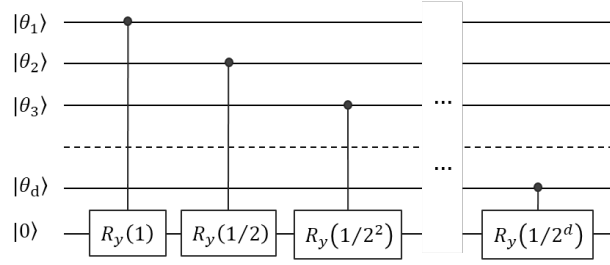


FIG. 9: The circuit for $c - R_y$.

D. Complexity

We now analyze the space and time complexity of the whole QAOP circuit. In one iteration Q of the QAOP circuit, the number of qubits required for preparing the initial quantum state $|\psi_{\mathbf{A}(0)}\rangle$ is $O(p + q) = O[\log_2(nk/\epsilon)]$. And phase estimation requires $O(\log_2 \kappa)$ qubits, where the condition number κ is usually taken as $\kappa = O(1/\epsilon)$. Therefore, the space complexity of phase estimation is $O(\log_2 \epsilon^{-1})$. Taking the ancilla qubits in the controlled rotation into account, the number of qubits in this stage is $O(d + b) = O(\log_2 \kappa) = O(\log_2 \epsilon^{-1})$. The number of qubits will not increase with the number of iterations of QAOP. To sum up, the total number of qubits required in the quantum circuit is $O[\log_2(nk/\epsilon)]$.

Now turning to the time consumption. In one iteration Q , the number of gates for the initial state preparing stage is $O(pq) = O[\log_2(nk) \log_2 \epsilon^{-1}]$. And phase estimation requires $O(b^2) = O[(\log_2 \kappa)^2] = O[(\log_2 \epsilon^{-1})^2]$ operations and two calls to the controlled-unitary black boxes. The number of quantum gates in the controlled rotation is $O[s'poly(d)] = O[s'poly(\log_2 \kappa)] = O[s'poly(\log_2 \epsilon^{-1})]$. The number of iteration Q is s , to sum up, the total time complexity of the QAOP circuit is $O[ss' \log_2(nk) poly(\log_2 \epsilon^{-1})]$.

In summary, $O[\log_2(nk/\epsilon)]$ space and $O[\log_2(nk)poly(\log_2\epsilon^{-1})]$ elementary operations allow us to implement the QAOP circuit with fidelity at least $1 - \epsilon$, when the number of iterations of the QAOP algorithm s and the number of Newton's iteration s' are both small constant numbers.

V. CONCLUSIONS

We have shown that the proposed quantum algorithm QAOP can be used to speed up the learning process of an important dimensionality reduction algorithm in pattern recognition and machine learning. We firstly reformulated the original AOP algorithm, therefore the quantum-classical interactions during the quantum algorithm can be omitted. We then proposed the QAOP algorithm and investigated the quantum circuits for solving the QAOP algorithm. The detailed quantum circuits for preparing the input quantum state and the circuits for solving the controlled rotation are presented. The space and time complexity of the quantum circuit shows that the number of the qubits and gates required are $O[\log_2(nk/\epsilon)]$ and $O[\log_2(nk)poly(\log_2\epsilon^{-1})]$, respectively. The time consumption shows The QAOP algorithm and QAOP circuit for dimensionality reduction may motivate to conduct new investigations in quantum machine learning.

Appendix A: Proof of theorem 1

Proof: According to Eq. (7), we have the initialization of $\mathbf{A}^{(0)} = \sum_{j=1}^k \beta_j^{(0)} |\mathbf{u}_j\rangle \langle \mathbf{j}|$, where $\beta_j^{(0)} = 1$ for all j 's. Obviously, $\langle \mathbf{u}_j | \mathbf{u}_{j'} \rangle = 0$ when $j \neq j'$ and $\langle \mathbf{u}_j | \mathbf{u}_{j'} \rangle = 1$ when $j = j'$. In the following, we simplify $\mathbf{A}^{(0)}$ as \mathbf{A} . Therefore, we have

$$\begin{aligned} \tilde{\mathbf{X}}^T \mathbf{A} &= \left(\sum_{j=1}^r \sigma_j |\mathbf{v}_j\rangle \langle \mathbf{u}_j| \right) \left(\sum_{j'=1}^k \beta_{j'}^{(0)} |\mathbf{u}_{j'}\rangle \langle \mathbf{j}'| \right) \\ &= \sum_{j=1}^k \sum_{j'=1}^k \sigma_j \beta_{j'}^{(0)} |\mathbf{v}_j\rangle \langle \mathbf{u}_j | \mathbf{u}_{j'} \rangle \langle \mathbf{j}'| = \sum_{j=1}^k \sigma_j \beta_j^{(0)} |\mathbf{v}_j\rangle \langle \mathbf{j}| \end{aligned} \tag{A1}$$

and $\tilde{\mathbf{X}}^T \mathbf{A} \mathbf{A}^T \tilde{\mathbf{X}} = \sum_{j=1}^k \left(\sigma_j \beta_j^{(0)} \right)^2 |\mathbf{v}_j\rangle \langle \mathbf{v}_j|$. According to the Eq. (5), we have:

$$\begin{aligned}
\mathbf{B} &= \left(\tilde{\mathbf{X}}^T \mathbf{A} \mathbf{A}^T \tilde{\mathbf{X}} + \lambda_2 \mathbf{I} \right)^{-1} \tilde{\mathbf{X}}^T \mathbf{A} \\
&= \left(\sum_{j=1}^k \left(\sigma_j \beta_j^{(0)} \right)^2 |\mathbf{v}_j\rangle \langle \mathbf{v}_j| + \lambda_2 \sum_{j=1}^r |\mathbf{v}_j\rangle \langle \mathbf{v}_j| \right)^{-1} \left(\sum_{j'=1}^k \sigma_{j'} \beta_{j'}^{(0)} |\mathbf{v}_{j'}\rangle \langle \mathbf{j}'| \right) \\
&= \left(\sum_{j=1}^k \left(\left(\sigma_j \beta_j^{(0)} \right)^2 + \lambda_2 \right) |\mathbf{v}_j\rangle \langle \mathbf{v}_j| \right)^{-1} \left(\sum_{j'=1}^k \sigma_{j'} \beta_{j'}^{(0)} |\mathbf{v}_{j'}\rangle \langle \mathbf{j}'| \right) \\
&= \left(\sum_{j=1}^k \left(\left(\sigma_j \beta_j^{(0)} \right)^2 + \lambda_2 \right)^{-1} |\mathbf{v}_j\rangle \langle \mathbf{v}_j| \right) \left(\sum_{j'=1}^k \sigma_{j'} \beta_{j'}^{(0)} |\mathbf{v}_{j'}\rangle \langle \mathbf{j}'| \right) \\
&= \sum_{j=1}^k \sum_{j'=1}^k \left(\left(\sigma_j \beta_j^{(0)} \right)^2 + \lambda_2 \right)^{-1} \sigma_{j'} \beta_{j'}^{(0)} |\mathbf{v}_j\rangle \langle \mathbf{v}_j| \mathbf{v}_{j'}\rangle \langle \mathbf{j}'| \\
&= \sum_{j=1}^k \frac{\sigma_j \beta_j^{(0)}}{\left(\sigma_j \beta_j^{(0)} \right)^2 + \lambda_2} |\mathbf{v}_j\rangle \langle \mathbf{j}|
\end{aligned} \tag{A2}$$

Similarly, we have

$$\tilde{\mathbf{X}} \mathbf{B} = \left(\sum_{j=1}^k \sigma_j |\mathbf{u}_j\rangle \langle \mathbf{v}_j| \right) \left(\sum_{j'=1}^k \frac{\sigma_{j'} \beta_{j'}^{(0)}}{\left(\sigma_{j'} \beta_{j'}^{(0)} \right)^2 + \lambda_2} |\mathbf{v}_{j'}\rangle \langle \mathbf{j}'| \right) = \sum_{j=1}^k \frac{\sigma_j^2 \beta_j^{(0)}}{\left(\sigma_j \beta_j^{(0)} \right)^2 + \lambda_2} |\mathbf{u}_j\rangle \langle \mathbf{j}|
\tag{A3}$$

$$\text{and } \tilde{\mathbf{X}} \mathbf{B} \mathbf{B}^T \tilde{\mathbf{X}}^T = \sum_{j=1}^k \left(\frac{\sigma_j^2 \beta_j^{(0)}}{\left(\sigma_j \beta_j^{(0)} \right)^2 + \lambda_2} \right)^2 |\mathbf{u}_j\rangle \langle \mathbf{u}_j|.$$

According to the Eq. (6), we have:

$$\begin{aligned}
\mathbf{A}^{(1)} &= \left(\tilde{\mathbf{X}} \mathbf{B} \mathbf{B}^T \tilde{\mathbf{X}}^T \right)^{-1} \tilde{\mathbf{X}} \mathbf{B} \\
&= \left(\sum_{j=1}^k \left(\frac{\left(\sigma_j \beta_j^{(0)} \right)^2 + \lambda_2}{\sigma_j^2 \beta_j^{(0)}} \right)^2 |\mathbf{u}_j\rangle \langle \mathbf{u}_j| \right) \left(\sum_{j'=1}^k \frac{\sigma_{j'}^2 \beta_{j'}^{(0)}}{\left(\sigma_{j'} \beta_{j'}^{(0)} \right)^2 + \lambda_2} |\mathbf{u}_{j'}\rangle \langle \mathbf{j}'| \right) \\
&= \sum_{j=1}^k \sum_{j'=1}^k \left(\frac{\left(\sigma_j \beta_j^{(0)} \right)^2 + \lambda_2}{\sigma_j^2 \beta_j^{(0)}} \right)^2 \frac{\sigma_{j'}^2 \beta_{j'}^{(0)}}{\left(\sigma_{j'} \beta_{j'}^{(0)} \right)^2 + \lambda_2} |\mathbf{u}_j\rangle \langle \mathbf{u}_j| \mathbf{u}_{j'}\rangle \langle \mathbf{j}'| \\
&= \sum_{j=1}^k \frac{\left(\sigma_j \beta_j^{(0)} \right)^2 + \lambda_2}{\sigma_j^2 \beta_j^{(0)}} |\mathbf{u}_j\rangle \langle \mathbf{j}|
\end{aligned} \tag{A4}$$

Eq. (A4) shows that after one iteration of the algorithm, only the singular values of \mathbf{A} are updated while the singular vectors stay the same. Therefore, for i -th iteration of the algorithm, let $\mathbf{A}^{(i)} = \sum_{j=1}^k \beta_j^{(i)} |\mathbf{u}_j\rangle \langle \mathbf{j}|$, where $\beta_j^{(0)} = 1$ for all j 's according to Eq. (7). We can easily have $\beta_j^{(i)} = \frac{(\sigma_j \beta_j^{(i-1)})^2 + \lambda_2}{\sigma_j^2 \beta_j^{(i-1)}}$.

□

ACKNOWLEDGMENTS

This work was supported by the Funding of National Natural Science Foundation of China (Grants No. 61571226 and No.61701229).

-
- [1] P. Chen, L. Jiao, F. Liu, J. Zhao, Z. Zhao, and S. Liu, *Pattern Recognition* **61**, 361 (2017).
 - [2] R. Wang, F. Nie, R. Hong, X. Chang, X. Yang, and W. Yu, *IEEE Transactions on Image Processing* **26**, 5019 (2017).
 - [3] Y. Song, Q. Li, H. Huang, D. Feng, M. Chen, and W. Cai, *IEEE Transactions on Medical Imaging* **36**, 1636 (2017).
 - [4] Y. Song, Q. Li, H. Huang, D. Feng, M. Chen, and W. Cai, *IEEE Transactions on Cybernetics* **99**, 1 (2018).
 - [5] X. He, C. Zhang, L. Zhang, and X. Li, *IEEE Transactions on Pattern analysis and machine intelligence* **38**, 1009 (2016).
 - [6] B. C. M., *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag New York, Inc., 2006).
 - [7] T. J. B, S. V. D, and L. J. C., *Science* **290**, 2319 (2000).
 - [8] S. Roweis and L. Saul, *Science* **290**, 2323 (2000).
 - [9] M. Belkin and P. Niyogi, in *International Conference on Neural Information Processing Systems: Natural and Synthetic* (2001) pp. 585–591.
 - [10] J. Adcock, E. Allen, M. Day, S. Frick, J. Hinchliff, M. Johnson, S. Morleyshort, S. Pallister, A. Price, and S. Stanisic, *arXiv:1512.02900* (2015).
 - [11] S. Lloyd, M. Mohseni, and P. Rebentrost, *Nature Physics* **10**, 108 (2014).
 - [12] I. Cong and L. Duan, *New Journal of Physics* **18**, 073011 (2016).

- [13] P. Wittek, *Quantum machine learning what quantum computing means to data mining* (Elsevier (Singapore) Pte Ltd., 2014).
- [14] M. Schuld, I. Sinayskiy, and F. Petruccione, Phys.Rev.A **94**, 022342 (2016).
- [15] B. Duan, J. Yuan, Y. Liu, and D. Li, Phys. Rev. A. **96**, 032301 (2017).
- [16] M. Schuld, I. Sinayskiy, and F. Petruccione, arXiv:1601.07823 (2016).
- [17] P. Kaye and M. Mosca, in *Proceedings of the International Conference on Quantum Information, OSA CDROM (Optical Society of America (2002) p. 28.*
- [18] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 160501 (2008).
- [19] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010) pp. 1–59.
- [20] V. V. Vedral, A. Barenco, and A. Ekert, Phys. Rev. A. **54**, 147 (1996).
- [21] A. Parent, M. Roetteler, and M. Mosca, arXiv preprint (2017).