PAPER

Quantum algorithm for neighborhood preserving embedding

Chin. Phys. B **31**, 060304 (2022)

Quantum algorithm for neighborhood preserving embedding

Shi-Jie Pan(潘世杰)[1,2], Lin-Chun Wan(万林春)[1], Hai-Ling Liu(刘海玲)[1], Yu-Sen Wu(吴宇森)[1],
Su-Juan Qin(秦素娟)[1], Qiao-Yan Wen(温巧燕)[1], and Fei Gao(高飞)[1,†]

[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing* 100876*, China*
[2] *State Key Laboratory of Cryptology, P.O. Box* 5159*, Beijing* 100878*, China*

(Received 15 November 2021; revised manuscript received 29 January 2022; accepted manuscript online 7 February 2022)

Neighborhood preserving embedding (NPE) is an important linear dimensionality reduction technique that aims at preserving the local manifold structure. NPE contains three steps, *i.e.*, finding the nearest neighbors of each data point, constructing the weight matrix, and obtaining the transformation matrix. Liang *et al.* proposed a variational quantum algorithm (VQA) for NPE [*Phys. Rev. A* **101** 032323 (2020)]. The algorithm consists of three quantum sub-algorithms, corresponding to the three steps of NPE, and was expected to have an exponential speedup on the dimensionality n. However, the algorithm has two disadvantages: (i) It is not known how to efficiently obtain the input of the third sub-algorithm from the output of the second one. (ii) Its complexity cannot be rigorously analyzed because the third sub-algorithm in it is a VQA. In this paper, we propose a complete quantum algorithm for NPE, in which we redesign the three sub-algorithms and give a rigorous complexity analysis. It is shown that our algorithm can achieve a polynomial speedup on the number of data points m and an exponential speedup on the dimensionality n under certain conditions over the classical NPE algorithm, and achieve a significant speedup compared to Liang *et al.*'s algorithm even without considering the complexity of the VQA.

Keywords: quantum algorithm, quantum machine learning, amplitude amplification

PACS: 03.67.Ac, 03.67.Lx **DOI:** 10.1088/1674-1056/ac523a

1. Introduction

Quantum computing theoretically demonstrates its computational advantages in solving certain problems compared with classical computing, such as the problem of factoring integers,[1] unstructured data searching problem,[2] and matrix computation problems.[3–5] In recent years, quantum machine learning has received widespread attention as a method that successfully combines classical machine learning with quantum physics. An important direction of quantum machine learning is to design quantum algorithms to accelerate classical machine learning, including data classification,[6–9] linear regression,[10–14] association rules mining,[15] and anomaly detection.[16]

Dimensionality reduction (DR) is an important part of machine learning, which aims to reduce the dimensionality of the training data set while preserving the structure information of the data points as well as possible. The DR algorithm often serves as a preprocessing step in data mining and machine learning to reduce the time complexity of the algorithm and avoid a problem called "curse of dimensionality".[17] Generally, The DR algorithms can be classified into two categories: the linear one and the nonlinear one. The most widely used linear DR algorithms include principal component analysis (PCA),[18] linear discriminant analysis (LDA),[19] and neighborhood preserving embedding (NPE),[20] while the typical nonlinear DR algorithm is locally linear embedding (LLE).[21] Here, we focus on NPE which can be regarded as the linear approximation of LLE. Unlike PCA that tries to preserve the global Euclidean structure, NPE aims at preserving the local manifold structure. Furthermore, NPE has a closed-form solution. Similar to other DR algorithms, NPE requires a large amount of computational resources in the big-data scenario because of its high complexity.

In recent years, some researchers successfully combined DR algorithms with quantum techniques and obtained various degrees of speedups. Lloyd *et al.*[22] proposed a quantum PCA algorithm to reveal the large eigenvectors in quantum form of an unknown low-rank density matrix, which achieves an exponential speedup on the dimension of the training data. Latter, Yu *et al.*[23] proposed a quantum algorithm that compresses training data based on PCA, and achieves an exponential speedup on the dimension over the classical algorithm. Cong *et al.*[8] proposed a quantum LDA algorithm for classification with exponential speedups on the scales of the training data over the classical algorithm. Besides, there are some other quantum DR algorithms, including quantum A-optimal projection,[24,25] quantum kernel PCA[26] and quantum spectral regression.[27,28]

For NPE, Liang *et al.*[29] proposed a variational quantum algorithm (VQA), called VQNPE, and expected to achieve an exponential speedup on dimensionality n. NPE contains three steps, *i.e.*, finding the nearest neighbors of each data point, constructing the weight matrix, and obtaining the transformation matrix A. VQNPE includes three quantum sub-algorithms

[†] Corresponding author. E-mail: gaof@bupt.edu.cn

with a VQA in the third sub-algorithm, corresponding to the three steps of NPE. However, VQNPE has two drawbacks: (i) The algorithm is incomplete. As the authors pointed out, it is not known how to obtain the input of the third sub-algorithm from the output of the second one. (ii) It lacks a provable quantum advantage. Since the advantage of VQA has not been proved rigorously yet (generally, we say that VQA has potential advantage[30,31]), it is hard to examine the speedups of the third sub-algorithm.

In this paper, we propose a complete quantum NPE algorithm with rigorous complexity analysis. Our quantum algorithm also consists of three quantum sub-algorithms, corresponding to the three steps of the classical NPE. The first one is finding the neighbors of each data point by quantum amplitude estimation and quantum amplitude amplification. By storing the information of neighbors in a data structure of QRAM,[32,33] we obtain two oracles. With these oracles, the second one reveals the classical information of the weight matrix W column by column by quantum matrix inversion technique. In the third one, we use a quantum version of the spectral regression (SR) method to get the transformation matrix A. Specifically, we obtain the d (d is the dimension of the low dimensional space) bottom nonzero eigenvectors of the matrix $M=(I-W)^{\mathrm{T}}(I-W)$ at first, and then perform several times of the quantum ridge regression algorithm to obtain A. As a conclusion, under certain conditions, our algorithm has a polynomial speedup on the number of data points m and exponential speedup on the dimension of the data points n over the classical NPE algorithm, and has a significant speedup compared with VQNPE.

The rest of this paper is organized as follows. In Section 2, we review the classical NPE algorithm. In Section 3, we propose our quantum NPE algorithm and analyze the complexity. Specifically, in Subsection 3.1, we propose a quantum algorithm to find the nearest neighbors of each data point and analyze the complexity. In Subsection 3.2, we propose a quantum algorithm to obtain the information of the weight matrix W and analyze the complexity. The quantum algorithm for computing the transformation matrix A is proposed in Subsection 3.3, together with the complexity analysis. The algorithm procedures and the complexity is concluded in Subsection 3.4, along with a comparison with VQNPE. The conclusion is given in Section 4.

2. Review of the classical NPE

In this section, we briefly review the classical NPE.[20,21,34]

Suppose $\boldsymbol{X}=(\boldsymbol{x}_0,\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{m-1})^{\mathrm{T}}$ is a data matrix with dimension $m\times n$, where n is the dimension of $\boldsymbol{x}_i$ and m is the number of data points. The objective of NPE is to find a matrix A (called transformation matrix) embedding the data matrix into a low-dimensional space (assume the embedding results is $\boldsymbol{y}_0,\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{m-1}$, $\boldsymbol{y}_i\in\mathbb{R}^d$ and $d\ll n$, we have $\boldsymbol{y}_i=A^{\mathrm{T}}\boldsymbol{x}_i$, $A\in\mathbb{R}^{n\times d}$) that the linear relation between each data point and its nearest neighbors is best preserved. Specifically, suppose the nearest neighbors of $\boldsymbol{x}_i$ are $\boldsymbol{x}_j,\boldsymbol{x}_k$, and $\boldsymbol{x}_l$, then $\boldsymbol{x}_i$ can be reconstructed (or approximately reconstructed) by linear combination of $\boldsymbol{x}_j,\boldsymbol{x}_k,\boldsymbol{x}_l$, that is,

$$\boldsymbol{x}_i=W_{ij}\boldsymbol{x}_j+W_{ik}\boldsymbol{x}_k+W_{il}\boldsymbol{x}_l, \tag{1}$$

where W_{ij}, W_{ik}, and W_{il} are weights that summarize the contribution of $\boldsymbol{x}_j$, $\boldsymbol{x}_k$, and $\boldsymbol{x}_l$ to the reconstruction of $\boldsymbol{x}_i$. NPE trys to preserve the linear relations in Eq. (1) in the low-dimensional embedding.

NPE consists of the following three steps.

Step 1 Find the nearest neighbors of each data point. There are two most common techniques to find the nearest neighbors. One is k-nearest neighbors algorithm (kNN) with a fixed k, and the other is choosing neighbors within a ball of fixed radius r based on Euclidean distance for each data point.

Step 2 Construct the weight matrix $W\in\mathbb{R}^{m\times m}$, where the $(i+1)$-th row and $(j+1)$-th column element is W_{ij}. Suppose the set of the nearest neighbors of the data point $\boldsymbol{x}_i$ is denoted as Q_i, then the construction of W is to optimize the following objective function:

$$\begin{aligned}&\min_{W}\sum_{i=0}^{m-1}\left\|\boldsymbol{x}_i-\sum_{j\in Q_i}W_{ij}\boldsymbol{x}_j\right\|^2,\\ &\text{s.t.}\quad \sum_{j\in Q_i}W_{ij}=1.\end{aligned} \tag{2}$$

Note that the data point $\boldsymbol{x}_i$ is only reconstructed by its nearest neighbors, *i.e.*, the elements in Q_i. If $\boldsymbol{x}_j\notin Q_i$, we set $W_{ij}=0$. We should mention that $\|\bullet\|$ is the L_2 norm of a vector or the spectral norm of a matrix in this paper. The above optimization problem has a closed form solution. Let $C^{(i)}$ denote an $m\times m$ matrix related to $\boldsymbol{x}_i$, called neighborhood correlation matrix, where

$$C^{(i)}_{jk}=\begin{cases}(\boldsymbol{x}_i-\boldsymbol{x}_j)^{\mathrm{T}}(\boldsymbol{x}_i-\boldsymbol{x}_k), & j,k\in Q_i;\\ 0, & \text{otherwise.}\end{cases} \tag{3}$$

Assume the number of elements of Q_i is $k^{(i)}$ and $k^{(i)}\ll m$, then $C^{(i)}$ are low-rank matrices for $i\in\{0,1,\ldots,m-1\}$. Let $C^{(i)}=\sum_{j=0}^{k^{(i)}-1}\lambda_j^{(i)}\boldsymbol{u}_j^{(i)}\boldsymbol{u}_j^{(i)\dagger}$, then the pseudo inverse of $C^{(i)}$ is $[C^{(i)}]^{-1}=\sum_{\lambda_j^{(i)}\neq 0}\frac{1}{\lambda_j^{(i)}}\boldsymbol{u}_j^{(i)}\boldsymbol{u}_j^{(i)\dagger}$. Let W_i denotes the $(i+1)$-th row of matrix W, then the solution of the objective function is

$$W_i=\frac{[C^{(i)}]^{-1}\mathbf{1}}{\mathbf{1}^{\mathrm{T}}[C^{(i)}]^{-1}\mathbf{1}}, \tag{4}$$

where $\mathbf{1}=(1,1,\ldots,1)^{\mathrm{T}}$.

Step 3 Compute the transformation matrix A. To best preserve the linear relations in the low-dimensional space, the optimization problem is designed as follows:

$$\min_{A} \sum_{i=0}^{m-1} \left\| \boldsymbol{y}_i - \sum_{j\in Q_i} W_{ij}\boldsymbol{y}_j \right\|^2 ,$$
$$\text{s.t.} \quad \sum_{i=0}^{m-1} \boldsymbol{y}_i = \boldsymbol{0}, \quad \frac{1}{m}\sum_{i=0}^{m} \boldsymbol{y}_i\boldsymbol{y}_i^{\mathrm{T}} = I, \quad \boldsymbol{y}_i = A^{\mathrm{T}}\boldsymbol{x}_i, \tag{5}$$

where $\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{m-1}$ are the low-dimensional embeddings. The optimization problem can be minimized by solving the following generalized eigen-problem:

$$X^{\mathrm{T}}MX\boldsymbol{a} = \lambda X^{\mathrm{T}}X\boldsymbol{a}, \tag{6}$$

where M is a sparse matrix that equates $(I-W)^{\mathrm{T}}(I-W)$. Then the bottom d nonzero eigenvectors $\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{d-1}$ of the above eigen-problem with corresponding eigenvalues $0 < \lambda_0 \le \lambda_1 \le \ldots \le \lambda_{d-1}$ yield $A = (\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{d-1})$.

There are many different methods to solve the eigenvalue problem in Eq. (6). Here we use the method mentioned in Refs. [35,36], called spectral regression (SR) method. The eigenvalue problem in Eq. (6) can be solved by two steps according to the SR method. (I) Solve the following eigen-problem to get the bottom non-zero eigenvectors $\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{d-1}$:

$$M\boldsymbol{z} = \lambda \boldsymbol{z}. \tag{7}$$

(II) Find $\boldsymbol{a}_i$ that satisfies

$$\begin{aligned} \boldsymbol{a}_i &= \arg\min_{\boldsymbol{a}} \left(\sum_{j=1}^{m} (\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}_j - \boldsymbol{z}_{ij})^2 + \alpha\|\boldsymbol{a}\|^2 \right) \\ &= \left(X^{\mathrm{T}}X + \alpha I \right)^{-1} X^{\mathrm{T}}\boldsymbol{z}_i, \end{aligned} \tag{8}$$

where $\boldsymbol{z}_{ij}$ is the j element of $\boldsymbol{z}_i$, $\alpha \ge 0$ is a constant to control the penalty of the norm of $\boldsymbol{a}$.

As a conclusion, the detailed procedures of NPE are given in Algorithm 1.

Algorithm 1 The procedure of NPE

```
Input: The data set $X = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{m-1})^{\mathrm{T}}$;
Output: The transformation matrix $A = (\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{d-1})$;
Find the set of nearest neighbors $Q_i$ of each data point $i$;
Construct $C^{(i)}$ by Eq. (3) for $i = 0, 1, \ldots, m-1$;
State Obtain $W$ by Eq. (4);
Decompose the matrix $M = (I-W)^{\mathrm{T}}(I-W)$ to get the bottom $d$ nonzero eigenvectors $\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{d-1}$;
Compute $\boldsymbol{a}_i = \left(X^{\mathrm{T}}X + \alpha I\right)^{-1} X^{\mathrm{T}}\boldsymbol{z}_i$ for $i = 0, 1, \ldots, d-1$;
return $A$.
```

As for the time complexity of NPE algorithm, the procedure to find the k nearest neighbors of each data point has complexity $O(mn\log_2 k \log_2 m)$ by using BallTree.[27] The complexity to construct the weight matrix W is $O(mnk^3)$ (generally, $k \ll m$). And the procedure to get the transformation matrix A has complexity $O(dm^2)$. Thus the overall complexity of NPE algorithm is $O(mnk^3 + dm^2)$.

3. Quantum algorithm for NPE

In this section, we introduce our quantum algorithm for NPE. The quantum algorithm can be divided into three parts, corresponding to the three parts of the classical algorithm. We give a quantum algorithm to find the nearest neighbors algorithm in Subsection 3.1, a quantum algorithm to construct the weight matrix W in Subsection 3.2 and a quantum algorithm to compute the transformation matrix A in Subsection 3.3. In Subsection 3.4, we conclude the complexity of our quantum algorithm and make a comparison with VQNPE.

3.1. Quantum algorithm to find the nearest neighbors

Assume that the data matrix $X = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{m-1})^{\mathrm{T}}$ is stored in a structured QRAM[32,33] which allows the following mappings to be performed in time $O[\mathrm{polylog}(mn)]$ as given below:

$$\begin{aligned} O_X &: |i\rangle|j\rangle|0\rangle \to |i\rangle|j\rangle|X_{ij}\rangle, \\ U_X &: |i\rangle|0\rangle \to \frac{1}{\|X_{i\cdot}\|}\sum_{j=1}^{n} X_{ij}|i,j\rangle = |i\rangle|\boldsymbol{x}_i\rangle, \\ V_X &: |0\rangle|j\rangle \to \frac{1}{\|X\|_F}\sum_{i=1}^{m} \|X_{i\cdot}\||i,j\rangle, \end{aligned} \tag{9}$$

where $X_{i\cdot}$ is the i-th row of X, *i.e.*, $\boldsymbol{x}_i$.

In our quantum algorithm, we choose neighbors within a ball of fixed radius r based on Euclidean distance for each data point (our algorithm can also be generalized to kNN to get a similar speedup). The selection of r is important for the performance of this type of algorithm, but how to choose a suitable r is outside the scope of our discussion. Here we assume that r is constant that given in advance. Let $k^{(i)}$ denote the number of nearest neighbors of $\boldsymbol{x}_i$, the objective of our quantum nearest neighbors algorithm is to output the index j of all $\boldsymbol{x}_j \in Q_i$, where $Q_i = \{\boldsymbol{x}_j | \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \le r^2, j \ne i, j \in \{0, 1, \ldots, m-1\}\}$ for $i \in \{0, 1, \ldots, m-1\}$, $|Q_i| = k^{(i)}$.

3.1.1. Algorithm details

We adopt the quantum amplitude estimation[38] and amplitude amplification[2,38] to get the neighbors of $\boldsymbol{x}_i$. The algorithm can be decomposed into the following two stages:

Stage 1 Prepare the following quantum state by quantum amplitude estimation,

$$|\phi\rangle = \frac{1}{m}\sum_{i,j=0}^{m-1}|i\rangle|j\rangle|\sqrt{K/m^2}\rangle, \tag{10}$$

where K is the number of the pairs of points that satisfy $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \le r$.

Stage 2 Prepare quantum state $\sqrt{p}|\psi\rangle + \sqrt{1-p}|\psi^{\perp}\rangle$, $p > 1/2$ by quantum amplitude amplification, where

$$|\psi\rangle = \frac{1}{\sqrt{K}}\sum_{i=0}^{m-1}|i\rangle\sum_{\boldsymbol{x}_j\in Q_i}|j\rangle, \tag{11}$$

$|\psi^{\perp}\rangle$ is a quantum state that is orthogonal to to $|\psi\rangle$. Then by measuring the state in computational basis for several times, we could obtain the index j of the neighbors of $\boldsymbol{x}_i$ for $i = 0,1,\ldots,m-1$.

Here we list a lemma that will be used in our algorithm:

Lemma 1 (Ref. [39]) Assume that $U : U|i\rangle|0\rangle = |i\rangle|\boldsymbol{v}_i\rangle$ and $V : V|j\rangle|0\rangle = |j\rangle|\boldsymbol{c}_j\rangle$ can be performed in time T, and the norms of the vectors $\boldsymbol{v}_i$ and $\boldsymbol{c}_j$ are known. Let $d^2(\boldsymbol{v}_i,\boldsymbol{c}_j) = \|\boldsymbol{v}_i - \boldsymbol{c}_j\|^2$, then a quantum algorithm can compute

$$|i\rangle|j\rangle|0\rangle \mapsto |i\rangle|j\rangle|d^2(\boldsymbol{v}_i,\boldsymbol{c}_j)\rangle, \tag{12}$$

with probability at least $1-2\delta$ for any δ with complexity $O(\frac{\|\boldsymbol{v}_i\|\|\boldsymbol{c}_j\|T\log_2(1/\delta)}{\varepsilon})$, where ε is the error of $d^2(\boldsymbol{v}_i,\boldsymbol{c}_j)$.

We now detail the stage 1. We first prepare the state $\frac{1}{m}\sum_{i,j=0}^{m-1}|i\rangle|j\rangle$. According to Lemma 1, we can obtain the state $\frac{1}{m}\sum_{i,j=0}^{m-1}|i\rangle|j\rangle|\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2\rangle$ by a unitary (denotes as U_1) with complexity $O\left[\frac{(\max_i\|\boldsymbol{x}_i\|)^2T\log_2(1/\delta)}{\varepsilon_1}\right]$, where $T = O[\mathrm{polylog}(mn)]$ is the complexity of the mappings in Eq. (9), $1-2\delta$ is the successful probability and ε_1 is the error of the value of $\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2$ stored in the third register.

Then, let O_1 be the unitary that transforms $\frac{1}{m}\sum_{i,j=0}^{m-1}|i\rangle|j\rangle|\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2\rangle$ to the state

$$\frac{1}{m}\sum_{i=0}^{m-1}|i\rangle\left(\sum_{\boldsymbol{x}_j\notin Q_i}|j\rangle|\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2\rangle - \sum_{\boldsymbol{x}_j\in Q_i}|j\rangle|\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2\rangle\right),$$

thus $O = U_1^{-1}O_1U_1$ transforms $\frac{1}{m}\sum_{i,j=0}^{m-1}|i\rangle|j\rangle$ to

$$\frac{1}{m}\sum_{i=0}^{m-1}|i\rangle\left(\sum_{\boldsymbol{x}_j\notin Q_i}|j\rangle - \sum_{\boldsymbol{x}_j\in Q_i}|j\rangle\right) := \cos(\theta)|\psi'\rangle - \sin(\theta)|\psi\rangle, \tag{13}$$

where $\sin\theta = \sqrt{K/m^2}$,

$$|\psi'\rangle = \sqrt{\frac{1}{m^2-K}}\sum_{i=0}^{m-1}|i\rangle\sum_{\boldsymbol{x}_j\notin Q_i}|j\rangle,$$

$|\psi\rangle$ is defined as Eq. (11). We can perform quantum amplitude estimation on $\frac{1}{m}\sum_{i,j=0}^{m-1}|i\rangle|j\rangle$ with oracle O and Grover operator G, where

$$G = H^{\otimes 2\log_2 m}(2|0\rangle\langle 0|^{2\log_2 m} - I_{m^2\times m^2})H^{\otimes 2\log_2 m}O.$$

Since G has eigenvalues $\mathrm{e}^{\pm 2\mathrm{i}\theta}$ with corresponding eigenvectors $|\psi_{\pm}\rangle$ and

$$\frac{1}{m}\sum_{i,j=0}^{m-1}|i\rangle|j\rangle = \frac{\mathrm{e}^{\mathrm{i}\theta}|\psi_+\rangle + \mathrm{e}^{-\mathrm{i}\theta}|\psi_-\rangle}{\sqrt{2}}, \tag{14}$$

where $|\psi_{\pm}\rangle = \frac{|\psi'\rangle \mp \mathrm{i}|\psi\rangle}{\sqrt{2}}$, the state after quantum amplitude estimation is

$$\frac{\mathrm{e}^{\mathrm{i}\theta}|\psi_+\rangle|\frac{\theta}{\pi}\rangle + \mathrm{e}^{-\mathrm{i}\theta}|\psi_-\rangle|\frac{\pi-\theta}{\pi}\rangle}{\sqrt{2}}. \tag{15}$$

Finally, by multiplying the value of the second register of the above state by π[40] and then applying a sine function,[41] we obtain the state $|\phi\rangle = \frac{1}{m}\sum_{i,j=0}^{m-1}|i\rangle|j\rangle|\sqrt{K/m^2}\rangle$, the final output of stage 1.

For the stage 2, based on the output of stage 1, we can inverse $\sqrt{K/m^2}$[42] and then multiple $\pi/4$[40] to get the state $|\phi_1\rangle = \frac{1}{m}\sum_{i=0}^{m-1}|i\rangle\sum_{j=0}^{m-1}|j\rangle|t\rangle$, where $t = \lceil\frac{\pi}{4}\sqrt{\frac{m^2}{K}}\rceil$. The ceil operator can be approximated by truncation since the error of t is required to be bounded by $t/3$ (see Subsection 3.1.2). Then we apply quantum amplitude amplification to $|\phi_1\rangle$. Specifically, we apply t times of G operator, which controlled by the third register of $|\phi_1\rangle$, *i.e.*,

$$\frac{1}{m}|t\rangle\sum_{i=0}^{m-1}|i\rangle\sum_{j=0}^{m-1}|j\rangle \to \frac{1}{m}|t\rangle G^t\sum_{i=0}^{m-1}|i\rangle\sum_{j=0}^{m-1}|j\rangle. \tag{16}$$

Thus we can obtain the quantum state

$$|t\rangle\left(\sqrt{p}|\psi\rangle + \sqrt{1-p}|\psi^{\perp}\rangle\right), \tag{17}$$

where $|\psi^{\perp}\rangle$ is the quantum state that is orthogonal to $|\psi\rangle$. If the estimation of t is sufficiently precise, for example, within error $t/3$, we have $p > 1/2$. Finally, by discarding the first register, we could get $\sqrt{p}|\psi\rangle + \sqrt{1-p}|\psi^{\perp}\rangle$.

3.1.2. Complexity analysis

To guarantee $p > 1/2$ of Eq. (17), the error of t should be less than $t/3$. Since $t = \lceil(\pi/4)\sqrt{m^2/K}\rceil$, we could make the error of K no more than $\frac{1}{2}K$.

In stage 1, since O_1 can be implemented in $O[\mathrm{polylog}(mn)]$ and U_1 has complexity $O\left[\frac{(\max_i\|\boldsymbol{x}_i\|)^2T\log_2(1/\delta)}{\varepsilon_1}\right]$, the complexity of the oracle $O = U_1^{-1}O_1U_1$ is

$$T_{\mathrm{oracle}} = O\left[\frac{(\max_i\|\boldsymbol{x}_i\|)^2T\mathrm{polylog}(mn/\delta)}{\varepsilon_1}\right].$$

The number of queries of O in quantum amplitude estimation is $O(\sqrt{m^2K}/\varepsilon_K) = O(\sqrt{m^2/K})$, $\varepsilon_K = \frac{1}{2}K$ is the estimate error of K. The complexity of multiplication of π and the sine gate can be neglected, which is $O(\text{polylog}(\sqrt{m^2K}/\varepsilon_K))$. Assume that the number of neighbors of each point $\boldsymbol{x}_i$ is balanced, that is, $k^{(i)} = \Theta(k)$ for $i = 0,1,\ldots,m-1$, $k^{(i)}$ is the number of neighbors of $\boldsymbol{x}_i$ (we should mention that if we adopt the kNN algorithm, $k^{(i)} = k$). Thus $K = \sum_i k^{(i)} = \Theta(mk)$. Therefore, the time complexity of stage 1 is

$$T_{s1}^{(1)} = O\left[\frac{(\max_i \|\boldsymbol{x}_i\|)^2 T\sqrt{m}\,\text{polylog}(mn/\delta)}{\varepsilon_1\sqrt{k}}\right].$$

In stage 2, the complexity of inversion and multiplication can be neglected. t times of G is implemented in the quantum amplitude amplification step to obtain the state in Eq. (17), where $t = \lceil(\pi/4)\sqrt{m^2/K}\rceil = O(\sqrt{m/k})$. Since the error of t is less than $t/3$, we have $p > 1/2$. We denote the complexity to get the state in Eq. (17) as $T_{s2}^{(1)}$. To reveal all pairs (i,j) that satisfy $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \le r$, we should measure the state $|\psi\rangle$ for $O(K\log_2 K)$ times.[43]

The δ would transform to an error $O(\sqrt{m/k}\delta)$ in $|\psi\rangle$, let $\delta = O(\sqrt{k\varepsilon^2/m})$, then the final error of $|\psi\rangle$ is ε. The ε_1 is related to the actual data set and the choice of r, here we assume it to be a constant. Let $h = \max_i \|\boldsymbol{x}_i\|$, note that $T = O[\text{polylog}(mn)]$, the total complexity of this algorithm is

$$\begin{aligned} T^{(1)} &= O(K\log_2 K(T_{s1}^{(1)} + T_{s2}^{(1)})) \\ &= O\left[h^2 m^{3/2} k^{1/2}\text{polylog}(mn/\varepsilon)\right]. \end{aligned} \tag{18}$$

3.2. Quantum algorithm to obtain the weight matrix W

Since we have obtained the indexes of the neighbors of all the data points in the previous algorithm. These information can be represented as a matrix B with $B_{ij} = 1$ if $\boldsymbol{x}_j \in Q_i$ and $B_{ij} = 0$ if $\boldsymbol{x}_j \notin Q_i$. To facilitate quantum access in the subsequent algorithms, we store the matrix B in the structured QRAM mention in Eq. (9), thus the following two mappings

$$\begin{aligned} &U_B : |i\rangle|0\rangle \mapsto |i\rangle|B_i\rangle, \\ &V_B : |0\rangle|j\rangle \mapsto \frac{1}{\sqrt{\|B\|_F}}\sum_i \|B_i\||i\rangle|j\rangle \end{aligned} \tag{19}$$

can be implemented in time $O[\text{polylog}(mn)]$, where

$$|B_i\rangle = \frac{1}{\sqrt{k^{(i)}}}\sum_{\boldsymbol{x}_j \in Q_i} |j\rangle. \tag{20}$$

The size of the data structure is $O[K\log_2^2(mn)] = O[mk\log_2^2(mn)]$, and the time to store the matrix B in the data structure is $O[K\log_2^2(mn)] = O[mk\log_2^2(mn)]$. We should mention that the complexity to construct the data structure (that is, store B in the data structure) can be neglected, since the complexity to obtain the matrix B is much greater.

Let $\rho_{C^{(i)}}$ be a dense matrix that is proportional to $C^{(i)}$, that is $\rho_{C^{(i)}} \propto C^{(i)}$, for $i = 0,1,2,\ldots,m-1$. Then the weight matrix $W = (W_1, W_2, \ldots, W_m)$ with

$$W_i = \frac{[C^{(i)}]^{-1}\mathbf{1}}{\mathbf{1}^{\mathrm{T}}[C^{(i)}]^{-1}\mathbf{1}} \propto \frac{\rho_{C^{(i)}}^{-1}|B_i\rangle}{\|\rho_{C^{(i)}}^{-1}|B_i\rangle\|} := |W_i\rangle. \tag{21}$$

To obtain each W_i, we give a quantum algorithm to prepare a state which is a purification of $\rho_{C^{(i)}}$ first. Then by means of this quantum algorithm, we can perform matrix inversion of $\rho_{C^{(i)}}$ on state $|B_i\rangle = (1/\sqrt{k^{(i)}})\sum_{\boldsymbol{x}_j \in Q_i} |j\rangle$ to obtain $|W_i\rangle$. Finally, the quantum state tomography[44–46] is used to reveal the information of $|W_i\rangle$. Since the state $|W_i\rangle$ is sparse, the quantum tomography step is efficient.

According to Eq. (3), we have

$$C^{(i)} = \sum_{\boldsymbol{x}_j, \boldsymbol{x}_k \in Q_i} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|\|\boldsymbol{x}_i - \boldsymbol{x}_k\|\langle \boldsymbol{x}_i - \boldsymbol{x}_j|\boldsymbol{x}_i - \boldsymbol{x}_k\rangle|j\rangle\langle k|,$$

where $|\boldsymbol{x}_i - \boldsymbol{x}_j\rangle$ denotes the quantum state which is proportional to the vector $\boldsymbol{x}_i - \boldsymbol{x}_j$, *i.e.*,

$$|\boldsymbol{x}_i - \boldsymbol{x}_j\rangle = \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}. \tag{22}$$

Let

$$|\psi^{(i)}\rangle = |i\rangle\frac{1}{\sqrt{c^{(i)}}}\sum_{\boldsymbol{x}_j \in Q_i} \|\boldsymbol{x}_i - \boldsymbol{x}_j\||j\rangle|\boldsymbol{x}_i - \boldsymbol{x}_j\rangle, \tag{23}$$

where $c^{(i)} = \sum_{\boldsymbol{x}_j \in Q_i} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$ is the normalized factor. $|\psi^{(i)}\rangle$ is a purification of $\rho_{C^{(i)}}$, since by taking partial trace of the first and third registers, we can obtain

$$\begin{aligned} \rho_{C^{(i)}} = &\frac{1}{c^{(i)}}\sum_{\boldsymbol{x}_j, \boldsymbol{x}_k \in Q_i} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|\|\boldsymbol{x}_i - \boldsymbol{x}_k\| \\ &\times \langle \boldsymbol{x}_i - \boldsymbol{x}_j|\boldsymbol{x}_i - \boldsymbol{x}_k\rangle|j\rangle\langle k|. \end{aligned} \tag{24}$$

3.2.1. Algorithm details

The algorithm to obtain each row of the weight matrix W, *i.e.*, W_i, can be decomposed to the following three stages:

Stage 1 Prepare the quantum state $|\psi^{(i)}\rangle$, thus we can obtain $\rho_{C^{(i)}}$.

Stage 2 Prepare quantum state $|W_i\rangle = \rho_{C^{(i)}}^{-1}|B_i\rangle$ by quantum matrix inversion technique.

Stage 3 Perform quantum state tomography on the state $|W_i\rangle$ to get the information of W_i.

Here we list a definition and three lemmas that will be used in our algorithm:

Definition 1 (Ref. [47]) An $(n+a)$-qubit unitary U is called an (α, a, ε) block-encoding of a matrix $A \in \mathbb{C}^{n\times n}$ if it satisfies

$$\|\alpha(\langle 0|^{\otimes a} \otimes I)U(|0\rangle^{\otimes a} \otimes I) - A\| \le \varepsilon, \tag{25}$$

where $\alpha > 0$.

Lemma 2 Given oracle O_1, O_2, O_1', and O_2' to access the vectors $\boldsymbol{x}_i$, $\boldsymbol{y}_i$ and the norm of the vectors in time $O(\text{polylog}(mn))$, *i.e.*,

$$O_1|i\rangle|0\rangle = |i\rangle|\boldsymbol{x}_i\rangle, \quad O_1'|i\rangle|0\rangle = |i\rangle|\|\boldsymbol{x}_i\|\rangle;$$

$$O_2|i\rangle|0\rangle = |i\rangle|\boldsymbol{y}_i\rangle, \quad O_2'|i\rangle|0\rangle = |i\rangle|\|\boldsymbol{y}_i\|\rangle,$$

there exists a quantum algorithm converts

$$\sum_{i,j=0}^{m-1}\sqrt{p_{ij}}|i\rangle|j\rangle \to \sum_{i,j=0}^{m-1}\sqrt{p_{ij}}|i\rangle|j\rangle|\boldsymbol{x}_i-\boldsymbol{y}_j\rangle, \tag{26}$$

with complexity $O(\frac{h}{\varepsilon_0}\text{polylog}(mn/\varepsilon))$, where $h = \max_i\{\|\boldsymbol{x}_i\|,\|\boldsymbol{y}_i\|\}$, ε is the error of the output state and $\varepsilon_0 = \min_{i,j}\|\boldsymbol{x}_i-\boldsymbol{y}_j\|$.

Proof see Appendix A.

Lemma 3 (Refs. [47,48]) Let G be an $(n+s)$-qubit unitary that generate ρ by tracing out the ancillary register, that is,

$$G|0\rangle = |G\rangle = \sum_j \sqrt{a_i}|j\rangle_1|\chi_j\rangle_2,$$
$$\rho = \text{Tr}(|G\rangle\langle G|)_1 = \sum_j a_i|\chi_j\rangle_2\langle\chi_j|_2.$$

Let S be a swap gate between register 2 and an ancillary system, *i.e*. register 3, then $(G^\dagger\otimes I_3)(I_1\otimes S_{2,3})(G\otimes I_3)$ is a $(1,n+s,0)$ block-encoding of ρ.

Lemma 4 (Ref. [49]) Let A be an $n\times n$ Hermitian matrix with non-zero eigenvalues lying in $[-1,-1/\kappa]\bigcup[1/\kappa,1]$, $\kappa\geq 2$. Assume that we have a unitary U which is an (α,a,δ) block-encoding of A that can be implemented in time $O(T_U)$, where $\delta = O(\varepsilon/(\kappa^2\log_2^3\frac{\kappa}{\varepsilon}))$. Also, assume that we can prepare the state $|b\rangle$ which spans the eigenvectors with non-zero eigenvalues of A in time $O(T_b)$. Then there is a quantum algorithm that output the quantum state $\frac{A^{-1}|b\rangle}{\|A^{-1}|b\rangle\|}$ with error ε in time

$$O\left(\kappa(\alpha(T_u+a)\log_2^2\left(\frac{\kappa}{\varepsilon}\right)+T_b)\log_2\kappa\right). \tag{27}$$

Lemma 5 (Refs. [44,45]) Assume that there is a quantum algorithm to prepare the quantum state $|\boldsymbol{x}\rangle = \sum_{i=0}^{d-1}x_i|i\rangle$ in time $O(T)$, then there is a quantum algorithm allows us to output a classical vector $\boldsymbol{x} = (x_0,x_1,\ldots,x_{d-1})^{\text{T}}$ that satisfies $\|\boldsymbol{x}-|\boldsymbol{x}\rangle\|\leq\delta$ in time $O(\frac{Td\log_2 d}{\delta^2})$ with probability at least $1-1/\text{poly}(d)$.

We now detail the stage 1. We first perform the U_B in Eq. (19) on the state $|i\rangle|0\rangle$ to get the state

$$|i\rangle\frac{1}{\sqrt{k^{(i)}}}\sum_{\boldsymbol{x}_j\in Q_i}|j\rangle. \tag{28}$$

According to Lemma 1, we can prepare the state

$$|i\rangle\frac{1}{\sqrt{k^{(i)}}}\sum_{\boldsymbol{x}_j\in Q_i}|j\rangle|\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2\rangle. \tag{29}$$

Using controlled rotation, we have

$$|i\rangle\frac{1}{\sqrt{k^{(i)}}}\sum_{\boldsymbol{x}_j\in Q_i}|j\rangle|\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2\rangle$$
$$\otimes\left(\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|}{r}|1\rangle+\sqrt{1-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{r^2}}|0\rangle\right). \tag{30}$$

Then by uncomputing the third register and measuring the last qubit to get $|1\rangle$, we can obtain the state

$$|i\rangle\frac{1}{\sqrt{c^{(i)}}}\sum_{\boldsymbol{x}_j\in Q_i}\|\boldsymbol{x}_i-\boldsymbol{x}_j\||j\rangle, \tag{31}$$

where $c^{(i)} = \sum_{\boldsymbol{x}_j\in Q_i}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2$ is the normalization factor.

Finally, according to Lemma 2, we can append $|\boldsymbol{x}_i-\boldsymbol{x}_j\rangle$ to the component marked by $|i\rangle|j\rangle$ of the state in Eq. (31) to get the state $|\psi^{(i)}\rangle$.

In stage 2, assume the unitary to prepare the state $|\psi^{(i)}\rangle$ is G'. Since $\text{Tr}(|\psi^{(i)}\rangle\langle\psi^{(i)}|)_{1,3} = \rho_{C^{(i)}}$, according to Lemma 3, we can obtain a $(1,2\log_2(m)+\log_2(n),0)$ block-encoding of $\rho_{C^{(i)}}$, that is, $(G'^\dagger\otimes I_4)(I_{1,3}\otimes S_{2,4})(G'\otimes I_4)$, S is a swap gate operated on register 2 and an ancillary register 4. According to Lemma 4, we can obtain $|W_i\rangle = \frac{\rho_{C^{(i)}}^{-1}|B_i\rangle}{\|\rho_{C^{(i)}}^{-1}|B_i\rangle\|}$.

In stage 3, we perform quantum tomography on $|W_i\rangle$ to get W_i. Although $|W_i\rangle$ is of dimension n, $|W_i\rangle$ is $k^{(i)}$ sparse and the positions of the non-zero amplitudes are known. We can perform a unitary to transform $|W_i\rangle$ to a state $|W_i'\rangle$ that only the first $k^{(i)}$ amplitudes are nonzero. $|W_i'\rangle$ can be regarded as a $\lceil\log_2(k^{(i)})\rceil$-dimension state, while the last $\log n-\lceil\log_2(k^{(i)})\rceil$ qubits are $|0\rangle$s. According to the Lemma 5, we can obtain a classical vector W_i'. It should be noted that W_i' satisfies $\sum_j W_{ij}' = 1$, thus renormalization is needed after quantum tomography. The renormalized vector W_i' is actually what we want since it contains all the information of W_i.

3.2.2. Complexity analysis

In stage 1, to prepare the state in Eq. (28), U_B is invoked for one time, thus the complexity is $O[\text{polylog}(mn)]$. According to Lemma 1, the complexity to prepare the state in Eq. (29) is

$$T_1 = O\left[\frac{(\max_i\|\boldsymbol{x}_i\|)^2 T\log_2(1/\delta)}{\varepsilon_1}\right],$$

where ε_1 is the error of $\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2$. The complexity of the controlled rotation can be neglected. Let $\varepsilon_0 = \min_{i,j}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|$, we have $\|\boldsymbol{x}_i-\boldsymbol{x}_j\|/r\geq\varepsilon_0/r$, where r is the fixed radius which can be regarded as a constant. The probability of obtaining $|1\rangle$ when measuring the last qubit of the state in Eq. (30) is

$$p(1) = \sum_{\boldsymbol{x}_j\in Q_i}\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{r^2k^{(i)}}\geq\varepsilon_0^2/r^2. \tag{32}$$

Using quantum amplitude amplification, $O(r/\varepsilon_0) = O(1/\varepsilon_0)$ times of repetition is enough to get the state in Eq. (31). The last step is to append $|\boldsymbol{x}_i-\boldsymbol{x}_j\rangle$ to the state in Eq. (31) to obtain $|\psi^{(i)}\rangle$. According to Lemma 2, it takes time $O(\frac{h}{\varepsilon_0}\text{polylog}(mn/\varepsilon))$, where $h = \max_i\|\boldsymbol{x}_i\|$. Let $\varepsilon_1 = O(\varepsilon^2\varepsilon_0^2)$, $\delta = O(\varepsilon_0\varepsilon)$, then the error of $|\psi^{(i)}\rangle$ is within $O(\varepsilon)$.

As a conclusion, the complexity of stage 1, *i.e.*, to prepare the state $|\psi^{(i)}\rangle$, is

$$T_{s1}^{(2)}=O\left[\frac{h^2}{\varepsilon^2\varepsilon_0^2}\mathrm{polylog}\left(\frac{mn}{\varepsilon\varepsilon_0}\right)\right].$$

For stage 2, since the error to prepare $|\psi^{(i)}\rangle$ is $O(\varepsilon)$, according to Lemma 3, we have a $(1,2\log_2 m+\log_2 n,\varepsilon)$ block-encoding of $\rho_{C^{(i)}}$. Note that $\rho_{C^{(i)}}$ is a sparse matrix that only $k^{(i)}$ rows and columns contains non-zero elements, thus the rank of $\rho_{C^{(i)}}$ is at most $k^{(i)}$. Without loss of generality, let the nonzero eigenvalues of $\rho_{C^{(i)}}$ be $\lambda_0\le\lambda_1\le\ldots\le\lambda_{k^{(i)}-1}$, we have $\sum_{j=0}^{k^{(i)}-1}\lambda_j=1$, $\max_i\lambda_i=\lambda_{k^{(i)}-1}\ge 1/k^{(i)}$. Assume the condition number of $C^{(i)}$ is $\kappa^{(i)}$, then $\lambda_j\in[\frac{1}{k^{(i)}\kappa^{(i)}},1]$. The state $|B_i\rangle$ can be prepared by oracle U_B in time $O(\mathrm{polylog}(mn))$. Thus according to Lemma 4, we can obtain $|W_i\rangle=\frac{\rho_{C^{(i)}}|B_i\rangle}{\|\rho_{C^{(i)}}|B_i\rangle\|}$ in time

$$T_2^{(2)}=O\left(\frac{h^2k^{(i)}\kappa^{(i)}}{\varepsilon^2\varepsilon_0^2}\mathrm{polylog}\left(\frac{mn}{\varepsilon\varepsilon_0}\right)\right).$$

In stage 3, according to Lemma 5, we can output the vector W_i' in $T_3^{(2)}=O(T_2^{(2)}k^{(i)}\log_2 k^{(i)}/\varepsilon^2)$ with probability at least $1-1/\mathrm{poly}(d)$, where ε is the error of vector W_i'. The complexity of renormalized can be neglected.

As a conclusion, the complexity to obtain the information of W is

$$T^{(2)}=O(mT_3^{(2)})=O\left(\frac{h^2mk_{\max}^2\kappa_{\max}}{\varepsilon^4\varepsilon_0^2}\mathrm{polylog}\left(\frac{mn}{\varepsilon\varepsilon_0}\right)\right),\quad(33)$$

where $k_{\max}=\max_i k^{(i)}$ and $\kappa_{\max}=\max_i\kappa^{(i)}$.

3.3. The quantum algorithm to compute the transformation matrix A

We have obtained the classical information of W in the above algorithm. Thus we can store the information of the matrix $D=I-W$ in a data structure that allows the following two mappings:

$$\begin{aligned}&U_D:|i\rangle|0\rangle\mapsto|i\rangle|D_i\rangle,\\&V_D:|0\rangle|j\rangle\mapsto\frac{1}{\sqrt{\|D\|_F}}\sum_i\|D_i\||i\rangle|j\rangle\end{aligned}\quad(34)$$

in time $O[\mathrm{polylog}(mn)]$, where $|D_i\rangle$ is proportional to the ith row of D. Note that W is a matrix of $K=\Theta(mk)$ nonzero elements and the diagonal elements are 0, the space and time complexity to construct the data structure of D are $O(mk\mathrm{polylog}(mk))$, the same as the matrix B.

Let $D=\sum_{j=0}^{m-1}\sigma_j|\boldsymbol{u}_j\rangle\langle\boldsymbol{v}_j|$, where $0\le\sigma_0\le\sigma_1\le\ldots\le\sigma_{m-1}$, then $M=D^{\mathrm{T}}D=\sum_j\sigma_j^2|\boldsymbol{v}_j\rangle\langle\boldsymbol{v}_j|$. According to Ref. [50], D is a matrix with rank no more than $m-1$. Without loss of generality, let the bottom d nonzero eigenvalue of M be $\sigma_{i_0}^2$ to $\sigma_{i_0+d-1}^2$ with corresponding eigenvectors $|\boldsymbol{v}_{i_0}\rangle,\ldots,|\boldsymbol{v}_{i_0+d-1}\rangle$.

To simplify the description of the quantum algorithm, we make a modification of the definition of the matrix X. If $m\le n$, we add $n-m$ zero rows to get an $n\times n$ matrix and vice versa. Here without loss of generality, we assume $m\le n$, then we get an $n\times n$ matrix X. We should mention that the data structure does not need to be modified.

Assume we have obtained $|\boldsymbol{v}_{i_0}\rangle$ to $|\boldsymbol{v}_{i_0+d-1}\rangle$ which are actually $\boldsymbol{z}_0,\boldsymbol{z}_1,\ldots,\boldsymbol{z}_{d-1}$ (see Eq. (7)). By appending $\log_2 n-\log_2 m$ zero states on $|\boldsymbol{v}_i\rangle$, we obtain a $\log_2 n$-qubit state $|\boldsymbol{v}_i^n\rangle=|0\rangle^{\otimes(\log_2 n-\log_2 m)}|\boldsymbol{v}_i\rangle$.

Let $X=\sum_i\gamma_i|\boldsymbol{u}_i^X\rangle\langle\boldsymbol{v}_i^X|$, $|\boldsymbol{v}_j^n\rangle=\sum_i\beta_i^{(j)}|\boldsymbol{u}_i^X\rangle$, then we have

$$\boldsymbol{a}_{j-i_0}=\left(X^{\mathrm{T}}X+\alpha I\right)^{-1}X^{\mathrm{T}}|\boldsymbol{v}_j^n\rangle=\sum_i\frac{\beta_i^{(j)}\gamma_i}{\gamma_i^2+\alpha}|\boldsymbol{v}_i^X\rangle.\quad(35)$$

Let

$$\begin{aligned}\overline{X}:&=\begin{bmatrix}0&X\\X^{\mathrm{T}}&0\end{bmatrix}=\begin{bmatrix}0&\sum_i\gamma_i|\boldsymbol{u}_i^X\rangle\langle\boldsymbol{v}_i^X|\\\sum_i\gamma_i|\boldsymbol{v}_i^X\rangle\langle\boldsymbol{u}_i^X|&0\end{bmatrix}\\&=\sum_i\pm\gamma_i|\psi_{i\pm}\rangle\langle\psi_{i\pm}|,\end{aligned}\quad(36)$$

where

$$|\psi_{i\pm}\rangle=\frac{1}{\sqrt{2}}\begin{bmatrix}|\boldsymbol{u}_i^X\rangle\\\pm|\boldsymbol{v}_i^X\rangle\end{bmatrix}=\frac{1}{\sqrt{2}}(|0,\boldsymbol{u}_i^X\rangle\pm|1,\boldsymbol{v}_i^X\rangle).\quad(37)$$

3.3.1. Algorithm details

The algorithm can be decomposed to the following stages:

1. Perform quantum singular value estimation (QSVE) to get the state

$$|\psi_1\rangle=\frac{1}{\sqrt{m}}\sum_{j=0}^{m-1}|\boldsymbol{v}_j\rangle|\boldsymbol{v}_j\rangle|\sigma_j\rangle.\quad(38)$$

2. Use the quantum algorithm for finding the minimum to find the d minimized nonzero value of σ_j, *i.e.*, $\sigma_{i_0},\sigma_{i_0+1},\ldots,\sigma_{i_0+d-1}$, and the corresponding $|\boldsymbol{v}_j\rangle$ for $j=i_0,i_0+1,\ldots,i_0+d-1$.

3. Perform quantum ridge regression to get $|\boldsymbol{a}_{j-i_0}\rangle\propto\boldsymbol{a}_{j-i_0}=\left(X^{\mathrm{T}}X+\alpha I\right)^{-1}X^{\mathrm{T}}\boldsymbol{v}_i$ for $j=i_0,i_0+1,\ldots,i_0+d-1$.

Here we list three lemmas that will be used in our algorithm:

Lemma 6 (QSVE[32,33]) Assume that an $m\times n$ matrix $D=\sum_j\sigma_j|\boldsymbol{u}_j\rangle\langle\boldsymbol{v}_j|$ is stored in a data structure that allows the two mappings showed in Eq. (34) in time $O(\mathrm{polylog}(mn))$. Let $\delta>0$ be the precision number. Then there is a quantum algorithm that transforms $\sum_j\beta_j|\boldsymbol{v}_j\rangle|0\rangle$ to $\sum_j\beta_j|\boldsymbol{v}_j\rangle|\overline{\sigma_j}\rangle$, where $\overline{\sigma_j}\in\sigma_j\pm\delta\|D\|_F$ with probability at least $1-1/\mathrm{poly}(n)$ in time $O(\frac{1}{\delta}\mathrm{polylog}(mn))$ for $j=0,1,\ldots,m-1$.

Lemma 7 (Ref. [49]) Let A be an $m\times n$ matrix stored in a data structure showed in Eq. (34), then there exists U_{R} and U_{L} that can be implemented in time $O(\mathrm{polylog}(mn)/\varepsilon)$ such that $U_{\mathrm{L}}^{\dagger}U_{\mathrm{R}}$ is a $(\|A\|_F,\lceil\log_2(m+n)\rceil,\varepsilon)$ block-encoding of $\overline{A}=\begin{bmatrix}0&A\\A^{\mathrm{T}}&0\end{bmatrix}$.

Lemma 8 (Ref. [49]) Suppose that U is an $(\alpha, a, \varepsilon/|2t|)$ block-encoding of H, then we can implement $\mathrm{e}^{\mathrm{i}Ht}$ with $O(|\alpha t| + \log_2(1/\varepsilon))$ query of U or its inverse and $O(a|\alpha t| + a\log_2(1/\varepsilon))$ two-qubit gates, where ε is the error of $\mathrm{e}^{\mathrm{i}Ht}$.

In stage 1, since we have no information about the state $|\boldsymbol{v}_j\rangle$ for $j = 0,1,\ldots,m-1$, we choose the state $|\psi_0\rangle = \frac{1}{\sqrt{m}}\sum_{j=0}^{m-1}|j\rangle|j\rangle$ to be the initial state, which can be written as uniform superposition of $|\boldsymbol{v}_j\rangle|\boldsymbol{v}_j\rangle$, *i.e.*,

$$|\psi_0\rangle = \frac{1}{\sqrt{m}}\sum_{j=0}^{m-1}|\boldsymbol{v}_j\rangle|\boldsymbol{v}_j\rangle. \tag{39}$$

Note that we have stored the matrix D in a suitable data structure and $D = \sum_{j=0}^{m-1}\sigma_j|\boldsymbol{u}_j\rangle\langle\boldsymbol{v}_j|$, according to QSVE (Lemma 6), we can obtain the state $|\psi_1\rangle$.

For stage 2, let U_2 be the unitary to prepare $|\psi_1\rangle$ from $|0\rangle$ and O_2 be the oracle to transform $|\psi_1\rangle$ to

$$\frac{1}{\sqrt{m}}\Bigg(\sum_{\sigma_j>v\,\mathrm{or}\,\sigma_j=0}|\boldsymbol{v}_j\rangle|\boldsymbol{v}_j\rangle|\sigma_j\rangle - \sum_{0<\sigma_j\leq v}|\boldsymbol{v}_j\rangle|\boldsymbol{v}_j\rangle|\sigma_j\rangle\Bigg), \tag{40}$$

where v is a constant. Given the U_2 and O_2, we can invoke the quantum algorithm for finding the minimum[51] to find the minimum nonzero value of σ_j, *i.e.*, σ_{i_0}. Assume that we have obtained i minimum nonzero values of σ_j for $j \in \{1,2,\ldots,m-1\}$, *i.e.*, σ_{i_0} to σ_{i_0+i-1}, we could make a small modification on O_2 to get an oracle $O_2^{(i)}$ that

$$O_2^{(i)} = \frac{1}{\sqrt{m}}\Bigg(\sum_{\sigma_j>v\,\mathrm{or}\,\sigma_j\leq\sigma_{i_0+i-1}}|\boldsymbol{v}_j\rangle|\boldsymbol{v}_j\rangle|\sigma_j\rangle - \sum_{\sigma_{i_0+i-1}<\sigma_j\leq v}|\boldsymbol{v}_j\rangle|\boldsymbol{v}_j\rangle|\sigma_j\rangle\Bigg).$$

Then we can perform the quantum algorithm for finding the minimum to find the $i+1$th minimum nonzero value of σ_j. Thus $O(d)$ times of the quantum algorithm for finding the minimum is enough to obtain the σ_{i_0} to σ_{i_0+d-1}. We should mention that when we get a $\sigma_j \in \{\sigma_{i_0},\sigma_{i_0+1},\ldots,\sigma_{i_0+d-1}\}$, we also get two samples of quantum state $|\boldsymbol{v}_j\rangle$.

For stage 3, since X is stored in the data structure, according to Lemma 7, we can implement an

$$(\|X\|_F, \lceil\log_2(2n)\rceil, \varepsilon_2)$$

block-encoding of $\overline{X}$. According to Lemma 8, we can implement $\mathrm{e}^{\mathrm{i}\overline{X}t}$. Then we can perform quantum ridge regression[13] to obtain $|\boldsymbol{a}_{j-i_0}\rangle$ for $j \in \{i_0, i_0+1,\ldots,i_0+d-1\}$. The algorithm proceeds as following steps:

1) Prepare the $\log_2(n)+1$ dimensional quantum state $|0,\boldsymbol{v}_j^n\rangle = \sum_i\beta_i^{(j)}|0,\boldsymbol{u}_i^X\rangle = \frac{1}{\sqrt{2}}\sum_i\beta_i^{(j)}(|\psi_{i+}\rangle + |\psi_{i-}\rangle)$ by expanding the quantum state $|\boldsymbol{v}_j\rangle$.

2) Perform quantum phase estimation on the state $|0,\boldsymbol{v}_j^n\rangle$ by simulating $\mathrm{e}^{\mathrm{i}\overline{X}t}$ to get the eigenvalues and eigenvectors of $\overline{X}$, *i.e.*, obtain the state

$$\frac{1}{\sqrt{2}}\sum_i\beta_i^{(j)}|\psi_{i\pm}\rangle|\pm\gamma_i\rangle. \tag{41}$$

3) Perform controlled rotation and uncompute the phase estimation to get the state

$$\frac{1}{\sqrt{2}}\sum_i\beta_i^{(j)}|\psi_{i\pm}\rangle\left(\frac{\pm C_1\gamma_i}{\gamma_i^2+\alpha}|0\rangle + \sqrt{1-\frac{C_1^2\gamma_i^2}{(\gamma_i^2+\alpha)^2}}|1\rangle\right),$$

where C_1 is a constant.

4) Measure the last qubit to get $|0\rangle$, and project the first register onto the $|\boldsymbol{v}_j^X\rangle$ part (*i.e.*, measure the first qubit of $|\psi_{i\pm}\rangle$ to get $|1\rangle$), we can obtain $|\boldsymbol{a}_{j-i_0}\rangle$,

$$|\boldsymbol{a}_{j-i_0}\rangle = \frac{1}{C}\sum_i\frac{\beta_i^{(j)}\gamma_i}{\gamma_i^2+\alpha}|\boldsymbol{v}_i^X\rangle, \tag{42}$$

where C is the normalized factor.

3.3.2. Complexity analysis

In stage 1, the preparation of $|\psi_0\rangle$ is of time $O(\log_2 m)$. Let the error of σ_j to be ε, according to Lemma 2, the complexity to get $|\psi_1\rangle$ is $O(\frac{\|D\|_F}{\varepsilon}\mathrm{polylog}(mn))$.

In stage 2, the O_2 (or O_2') and U_2 can be implemented in time $O(\mathrm{polylog}(m))$ and $O(\frac{\|D\|_F}{\varepsilon}\mathrm{polylog}(mn))$, respectively. The quantum algorithm for finding the minimum would output the minimum value with probability larger than $1/2$ with query complexity $O(\sqrt{m})$. We should mention that one query includes two U_2 and one O_2 (or O_2'). Thus to get σ_{i_0} to σ_{i_0+d-1} and $|\boldsymbol{v}_{i_0}\rangle$ to $|\boldsymbol{v}_{i_0+1-d}\rangle$, $O(d)$ times of the algorithm for finding the minimum is enough. The total complexity is $O(\frac{d\sqrt{m}\|D\|_F}{\varepsilon}\mathrm{polylog}(mn))$.

In stage 3, for the step 1, we can append several $|0\rangle$ to $|\boldsymbol{v}_j\rangle$ to get $|0,\boldsymbol{v}_j^n\rangle$. For the Step 2, an

$$(\|X\|_F, \lceil\log_2(2n)\rceil, \varepsilon_2)$$

block-encoding of $\overline{X}$ can be implemented in time $O(\mathrm{polylog}(mn)/\varepsilon_2)$. According to Lemma 8, we can simulate $e^{\mathrm{i}\overline{X}t}$ in time $O(\|X\|_F t\,\mathrm{polylog}(mn/\varepsilon_2))$, where error $\varepsilon_3 = 2t\varepsilon_2$. Let κ denote the condition number of X, to ensure the error of the final state $|\boldsymbol{a}_{j-i_0}\rangle$ is within ε, the maximum simulation time of the quantum phase estimation should be $t = O(\kappa/\varepsilon)$ and $\varepsilon_3 = \varepsilon/\log_2(\kappa/\varepsilon)$. Thus the complexity of the quantum phase estimation is $O(\frac{\|X\|_F\kappa}{\varepsilon}\mathrm{polylog}(mn/\varepsilon))$. The complexity of Step 3 is the same as Step 3. In Step 4, we could choose $C_1 = O(\max_i(\frac{\gamma_i}{\gamma_i^2+\alpha}))^{-1}$, thus have $\frac{C_1\gamma_i}{\gamma_i^2+\alpha} = O(1/\kappa)$.[13] $O(\kappa^2)$ repetitions are needed to get a $|0\rangle$ and it can be improved to $O(\kappa)$ repetitions by quantum amplitude amplification. The projection is success with probability $1/2$.

As a conclusion, the complexity to get $|\boldsymbol{a}_0\rangle, |\boldsymbol{a}_1\rangle, \ldots, |\boldsymbol{a}_{d-1}\rangle$ is

$$O\left(\kappa\frac{d\sqrt{m}\|D\|_F}{\varepsilon}\mathrm{polylog}(mn) + \frac{d\|X\|_F\kappa^2}{\varepsilon}\mathrm{polylog}\left(\frac{mn}{\varepsilon}\right)\right)$$

$$= O\left(d\left(\frac{\sqrt{m}\|D\|_F\,\kappa+\|X\|_F\,\kappa^2}{\varepsilon}\right)\mathrm{polylog}\left(\frac{mn}{\varepsilon}\right)\right).$$

Assume that $\|W\|_{\max}=O(1)$. Since W is sparse matrix and $D=I-W$, we have $\|D\|_F=O(\sqrt{mk})$. As for the $\|X\|_F$, since $h=\max_i\|\boldsymbol{x}_i\|$, $d\|X\|_F=O(\sqrt{hm})$. Thus the complexity of the algorithm is $O(\frac{d\sqrt{hk}m\kappa^2}{\varepsilon}\mathrm{polylog}(\frac{mn}{\varepsilon}))$.

3.4. The total complexity and discussion

The procedure of the quantum NPE algorithm can be summarized as follows:

Algorithm 2 The procedure of quantum NPE

```
Input: The data matrix $X$ is stored in a data structure;
Output: The quantum states $|\boldsymbol{a}_0\rangle,|\boldsymbol{a}_1\rangle,\ldots,|\boldsymbol{a}_{d-1}\rangle$ which represent each row of matrix $A$;
1: Prepare $\frac{1}{m}\sum_{i,j=0}^{m-1}|i\rangle|j\rangle|\sqrt{\frac{K}{m^2}}\rangle$;
2: Prepare $\frac{1}{\sqrt{K}}\sum_{i=0}^{m-1}|i\rangle\sum_{\boldsymbol{x}_j\in Q_i}|j\rangle$;
3: Measure the output in computational basis for several times to obtain the index $j$ of the neighbors of $\boldsymbol{x}_i$ for $i=0,1,\ldots,m-1$;
4: Construct oracle $U_B$ and $V_B$;
5: Prepare $|\psi^{(i)}\rangle$ to obtain $\rho_{C^{(i)}}$;
6: Prepare $|W_i\rangle=\rho_{C^{(i)}}^{-1}|B_i\rangle$ for $i=0,1,\ldots,m-1$;
7: Perform quantum state tomography on $|W_i\rangle$ to get the information of $W_i$ for $i=0,1,\ldots,m-1$;
8: Perform quantum singular value estimation to get $|\psi_1\rangle$;
9: Use the quantum algorithm for finding the minimum to find the bottom $d$ nonzero eigenvalues and eigenvectors of $M$;
10: Perform quantum ridge regression to get $|\boldsymbol{a}_j\rangle$ for $j=0,1,\ldots,d-1$.
11: return $|\boldsymbol{a}_0\rangle,|\boldsymbol{a}_1\rangle,\ldots,|\boldsymbol{a}_{d-1}\rangle$.
```

The quantum algorithm can be divided into three sub-algorithms and the complexity of each sub-algorithm can be seen in Table 1. The algorithms 1–3 in Table 1 are the quantum algorithm to find the nearest neighbors, the algorithm to obtain the weight matrix W and the algorithm for embedding, respectively. $h=\max_i\|\boldsymbol{x}_i\|$, $k=\Theta(k^{(i)})$, $k^{(i)}$ is the number of neighbors of $\boldsymbol{x}_i$, m is the number of training data points, n is the dimension of the data points, ε is the error of the algorithm, $\varepsilon_0=\min_{ij}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|$, $k_{\max}=\max_i k^{(i)}$, $\kappa_{\max}=\max_i\kappa^{(i)}$, $\kappa^{(i)}$ is the condition number of the neighborhood correlation matrix $C^{(i)}$, d is the dimension of the low-dimensional space, κ is the condition number of train data matrix X. Putting it all together, the complexity of the quantum NPE algorithm is

$$O\left((h^2m^{3/2}k^{1/2}+\frac{dh^2mk_{\max}^2\kappa_{\max}\kappa^2}{\varepsilon^4\varepsilon_0^2})\mathrm{polylog}\left(\frac{mn}{\varepsilon\varepsilon_0}\right)\right).$$

Table 1. The time complexity of the three sub-algorithms of the quantum NPE.

Sub-algorithm[a]	Time complexity
Algorithm 1	$O\left(h^2m^{3/2}k^{1/2}\mathrm{polylog}\left(\frac{mn}{\varepsilon}\right)\right)$
Algorithm 2	$O\left(\frac{h^2mk_{\max}^2\kappa_{\max}}{\varepsilon^4\varepsilon_0^2}\mathrm{polylog}\left(\frac{mn}{\varepsilon\varepsilon_0}\right)\right)$
Algorithm 3	$O\left(\frac{d\sqrt{hk}m\kappa^2}{\varepsilon}\mathrm{polylog}\left(\frac{mn}{\varepsilon}\right)\right)$

Since the classical algorithm have complexity $O(mnk^3+dm^2)$, our algorithm have a polynomial speedup on m and an exponential speedup on n when the factors $d,h,k_{\max},\kappa_{\max},\varepsilon,\varepsilon_0=O[\mathrm{polylog}(mn)]$. We should mention that the output of the quantum NPE algorithm is a matrix $A=(|\boldsymbol{a}_0\rangle,|\boldsymbol{a}_1\rangle,\ldots,|\boldsymbol{a}_{d-1}\rangle)$ with each column outputted as a quantum state.

Our algorithm has two advantages over VQNPE. (i) Our algorithm is complete while VQNPE is not. In Ref. [29], the authors pointed out that it is not known how to obtain the input of the third sub-algorithm from the output of the second sub-algorithm. Actually, if just consider the completeness, we can use a VQA technique shown in Ref. [52] after our second sub-algorithm. (ii) The complexity of our algorithm is less than the complexity of VQNPE, even without considering the complexity of the third sub-algorithm of VQNPE. Specifically, The complexity of the first sub-algorithm is $O(\frac{m^2}{\varepsilon^2}\log_2 n)$, and the complexity of the second sub-algorithm is $\Omega(\mathrm{poly}(n))$ (we should mention that the complexity showed here are different with the original paper, see Appendix B for details), while the total complexity of our algorithm is $O(m^{1.5}\mathrm{polylog}(mn))$ (only consider the main parameters). The advantage of our first sub-algorithm is mainly coming from the parallel estimation of the distance of each pair of data points. As for the second sub-algorithm, Liang *et al.* adopted the QSVD to get the $|W_i\rangle$. However, the eigenvalues of $\mathcal{A}_i$ are too small to satisfy the conditions to get an efficient algorithm, which causes the complexities to have polynomial dependence on n. We use a totally different algorithm to get the $|W_i\rangle$ and the complexity analysis shows that our algorithm has complexity polylogarithmic dependence on n. As for the third sub-algorithm, it is hard to exam the complexity of the VQA of VQNPE, while our sub-algorithm has a rigorous complexity analysis.

Recently, derived from the quantum-inspired algorithm proposed by Tang,[53] Chia *et al.*[54] introduced a framework for applying singular value transformation (SVT) for dequantizing quantum machine learning algorithms. They proved that in the QRAM data structure input model, the exponential speedups of quantum SVT can be broke by dequantization technique. We should mention that the dequantization technique is not suitable for NPE, although our third sub-algorithm use QSVE which is a type of SVT. Since $\|D\|_F = \sqrt{mk}$, if we adopt dequantization technique in our third sub-algorithm, the dependence on m would be much higher than m^2.

Also, we notice that Chen *et al.*[55] proved a lower bound of quantum ridge regression algorithm. However, since their objective is to return a classical vector as the output of ridge regression while our algorithm (the ridge regression algorithm in the third sub-algorithm) generates a solution vector as a quantum state, the lower bound would not influence the complexity bound of our algorithm.

4. Conclusion

In this paper, we proposed a complete quantum NPE algorithm with rigorous complexity analysis. It was showed that when $d, h, k_{\max}, \kappa_{\max}, \varepsilon, \varepsilon_0 = O[\mathrm{polylog}(mn)]$, our algorithm has exponential acceleration on n and polynomial acceleration on m over the classical NPE. Also, our algorithm has a significant speedup compared with VQNPE.

The Lemma 2 proposed an efficient method to append a quantum state generated by subtracting two vectors parallelly, which might have a wide range of applications in other quantum algorithms. Also, in the proof of the Lemma 2, we proposed a technique called parallel amplitude amplification, which may be of independent interest. We hope the techniques used in our algorithm could inspire more DR techniques to get a quantum advantage, especially the nonlinear DR techniques. We will explore the possibility in the future.

Acknowledgements

We thank Chao-Hua Yu for fruitful discussion and the anonymous referees for their helpful comments.

Project supported by the Fundamental Research Funds for the Central Universities (Grant No. 2019XD-A01) and the National Natural Science Foundation of China (Grant Nos. 61972048 and 61976024).

Appendix A: The proof of Lemma 2

Proof $|\boldsymbol{x}_i - \boldsymbol{y}_j\rangle$ is a quantum state that is proportional to vector $\boldsymbol{x}_i - \boldsymbol{y}_j$,

$$\begin{aligned} |\boldsymbol{x}_i - \boldsymbol{y}_j\rangle &= |\|\boldsymbol{x}_i\||\boldsymbol{x}_i\rangle - \|\boldsymbol{y}_i\||\boldsymbol{y}_i\rangle\rangle \quad (43) \\ &= \frac{\|\boldsymbol{x}_i\||\boldsymbol{x}_i\rangle - \|\boldsymbol{y}_i\||\boldsymbol{y}_i\rangle}{\|\|\boldsymbol{x}_i\||\boldsymbol{x}_i\rangle - \|\boldsymbol{y}_i\||\boldsymbol{y}_i\rangle\|}. \quad \text{(A1)} \end{aligned}$$

A1: Algorithm details

Let

$$|\psi\rangle := \frac{1}{\sqrt{m}} \sum_{i,j=0}^{m-1} \sqrt{p_{ij}} |i\rangle|j\rangle|\boldsymbol{x}_i - \boldsymbol{y}_j\rangle, \quad \text{(A2)}$$

the process to prepare $|\psi\rangle$ from $\frac{1}{\sqrt{m}} \sum_{i,j=0}^{m-1} \sqrt{p_{ij}}|i\rangle|j\rangle$ can be summarized as follows:

1. Given quantum state $\sum_{i,j=0}^{m-1} \sqrt{p_{ij}}|i\rangle|j\rangle$, prepare $\sum_{i,j=0}^{m-1} \sqrt{p_{ij}}|i\rangle|j\rangle|\|\boldsymbol{x}_i\|\rangle|\|\boldsymbol{y}_j\|\rangle$.

2. Prepare the following quantum state by controlled rotation,[3]

$$\sum_{i,j=0}^{m-1} \sqrt{p_{ij}}|i\rangle|j\rangle|\|\boldsymbol{x}_i\|\rangle|\|\boldsymbol{y}_j\|\rangle(\cos\theta_{ij}|0\rangle + \sin\theta_{ij}|1\rangle), \quad \text{(A3)}$$

where $\cos\theta_{ij} = \frac{\|\boldsymbol{x}_i\|}{\sqrt{\|\boldsymbol{x}_i\|^2 + \|\boldsymbol{y}_j\|^2}}$, and thus

$$\sin\theta_{ij} = \frac{\|\boldsymbol{y}_j\|}{\sqrt{\|\boldsymbol{x}_i\|^2 + \|\boldsymbol{y}_j\|^2}}.$$

3. Uncompute the third and the fourth registers, and then query the oracles to obtain the state

$$\sum_{i,j=0}^{m-1} \sqrt{p_{ij}}|i\rangle|j\rangle(\cos\theta_{ij}|0\rangle|\boldsymbol{x}_i\rangle + \sin\theta_{ij}|1\rangle|\boldsymbol{y}_j\rangle). \quad \text{(A4)}$$

4. Apply Hadamard gate to the third register to obtain

$$\begin{aligned} &\sum_{i,j=0}^{m-1} \sqrt{p_{ij}}|i\rangle|j\rangle\frac{1}{\sqrt{2}}\big[|0\rangle(\cos\theta_{ij}|\boldsymbol{x}_i\rangle + \sin\theta_{ij}|\boldsymbol{y}_j\rangle) \\ &+ |1\rangle(\cos\theta_{ij}|\boldsymbol{x}_i\rangle - \sin\theta_{ij}|\boldsymbol{y}_j\rangle)\big] \\ :=& \sum_{i,j=0}^{m-1} \sqrt{p_{ij}}|i\rangle|j\rangle(\cos\psi_{ij}|\phi_{ij}^+\rangle + \sin\psi_{ij}|\phi_{ij}^-\rangle), \quad \text{(A5)} \end{aligned}$$

where

$$\begin{aligned} \cos\psi_{ij}|\phi_{ij}^+\rangle &= \frac{1}{\sqrt{2}}|0\rangle(\cos\theta_{ij}|\boldsymbol{x}_i\rangle + \sin\theta_{ij}|\boldsymbol{y}_j\rangle), \\ \sin\psi_{ij}|\phi_{ij}^-\rangle &= \frac{1}{\sqrt{2}}|1\rangle(\cos\theta_{ij}|\boldsymbol{x}_i\rangle - \sin\theta_{ij}|\boldsymbol{y}_j\rangle). \end{aligned}$$

5. Perform parallel quantum amplitude amplification to get state

$$\sum_{i,j=0}^{m-1} \sqrt{p_{ij}}|i\rangle|j\rangle|\phi_{ij}^-\rangle = \sum_{i,j=0}^{m-1} \sqrt{p_{ij}}|i\rangle|j\rangle|1\rangle|\boldsymbol{x}_i - \boldsymbol{y}_j\rangle. \quad \text{(A6)}$$

6. Discard the third register, the state left is $\sum_{i,j=0}^{m-1} \sqrt{p_{ij}}|i\rangle|j\rangle|\boldsymbol{x}_i - \boldsymbol{y}_j\rangle$.

To make the Step 5 (parallel quantum amplitude amplification) more clear, we give details here. Let U_1 be the unitary that prepares the state in Eq. (A5) from quantum state $\sum_{i,j=0}^{m-1} \sqrt{p_{ij}}|i\rangle|j\rangle$, and O be the unitary that transforms the state in Eq. (A5) to

$$\sum_{i,j=0}^{m-1} \sqrt{p_{ij}}|i\rangle|j\rangle(\cos\psi_{ij}|\phi_{ij}^+\rangle - \sin\psi_{ij}|\phi_{ij}^-\rangle). \quad \text{(A7)}$$

We first perform a parallel quantum amplitude estimation[15] to obtain the amplitudes of the target states. And then we perform the fixed-point quantum search[56] parallelly to obtain the final state. We defined the Grover operator of the parallel quantum amplitude estimation as

$$U_1(I_{m^2\times m^2} \otimes (2|0\rangle\langle 0|^{\otimes(1+\log_2 n)} - I_{2n\times 2n}))U_1^{\dagger}O. \quad \text{(A8)}$$

The parallel quantum amplitude amplification consists of three steps:

Step 1 Perform quantum amplitude estimation on the quantum state in Eq. (A5) to get the estimated values of $|\sin\psi_{ij}| := \frac{1}{\sqrt{2}}\|\cos\theta_{ij}|\boldsymbol{x}_i\rangle - \sin\theta_{ij}|\boldsymbol{y}_i\rangle)\|$ parallelly for each $|i\rangle|j\rangle$, *i.e.*, obtain the state

$$\sum_{i,j=0}^{m-1}\sqrt{p_{ij}}|i\rangle|j\rangle(\cos\psi_{ij}|\phi_{ij}^{+}\rangle+\sin\psi_{ij}|\phi_{ij}^{-}\rangle)||\sin\psi_{ij}|\rangle.$$

Step 2 Let $L_{ij}=2\lceil\frac{\log_2(2/\delta')}{|\sin\psi_{ij}|}\rceil$, prepare the state

$$\sum_{i,j=0}^{m-1}\sqrt{p_{ij}}|i\rangle|j\rangle(\cos\psi_{ij}|\phi_{ij}^{+}\rangle+\sin\psi_{ij}|\phi_{ij}^{-}\rangle)|L_{ij}\rangle, \tag{A9}$$

where $\delta'>0$ is a parameter that related to the final error.

Step 3 Controlled by $|L_{ij}\rangle$ and $|i\rangle|j\rangle$, we perform $S_{L_{ij}} = G(\alpha_l,\beta_l)\ldots G(\alpha_1,\beta_1)=\prod_{k=1}^{l_{ij}}G(\alpha_k,\beta_k)$ on the third register of the above equation, where $l_{ij}=\lceil\frac{L_{ij}-1}{2}\rceil$, for all $k=1,2,\ldots,l$,

$$\alpha_k=-\beta_{l-k+1}=2\cot^{-1}\left(\tan(2\pi k/L_{ij})\sqrt{1-\gamma^2}\right),$$

$\gamma^{-1}=T_{1/L_{ij}}(1/\delta')$, $T_{\mathrm{L}}(x)=\cos(L\cos^{-1}(x))$ is the Lth Chebyshev polynomial of the first kind. We can obtain

$$\sum_{i,j=0}^{m-1}\sqrt{p_{ij}}|i\rangle|j\rangle|\widetilde{\phi_{ij}^{-}}\rangle,$$

where $\|\langle\widetilde{\phi_{ij}^{-}}|\phi_{ij}^{-}\rangle\|^2\geq 1-\delta'^2$.

A2: Complexity analysis

In Step 1, two times of queries are invoked. The complexity of Step 2 can be neglected. In Step 3, four queries are invoked. The Hadamard gate in Step 4 is of complexity $O(1)$.

As for the Step 5, we should amplify the amplitudes $\sin\psi_{ij}$ that satisfies

$$\begin{aligned}|\sin\psi_{ij}| &= \|\frac{1}{\sqrt{2}}|1\rangle(\cos\theta_{ij}|\boldsymbol{x}_i\rangle-\sin\theta_{ij}|\boldsymbol{y}_j\rangle)\|\\ &=\|\frac{1}{\sqrt{2}}\frac{\boldsymbol{x}_i-\boldsymbol{y}_j}{\sqrt{\|\boldsymbol{x}_i\|^2+\|\boldsymbol{y}_j\|^2}})\|\\ &\geq\frac{\varepsilon_0}{2h},\end{aligned} \tag{A10}$$

where $\varepsilon_0=\min_{i,j}\|\boldsymbol{x}_i-\boldsymbol{y}_j\|$, $h=\max_i\{\|\boldsymbol{x}_i\|,\|\boldsymbol{y}_i\|\}$. The complexity of one query of the Grover operator in Eq. (A8) is $O(\text{polylog}(mn))$. According to Ref. [56], we should ensure that $L_{ij}\geq\frac{\log_2(2/\delta')}{|\sin\psi_{ij}|}$. We could estimate $|\sin\psi_{ij}|$ within error $\frac{1}{2}|\sin\psi_{ij}|$, then choose $L_{ij}=2\lceil\frac{\log_2(2/\delta')}{|\sin\psi_{ij}|}\rceil$ to ensure $L_{ij}\geq\frac{\log_2(2/\delta')}{|\sin\psi_{ij}|}$. With the error $\frac{1}{2}|\sin\psi_{ij}|$, the complexity of the parallel quantum amplitude estimation is $O(\frac{h}{\varepsilon_0}\text{polylog}(mn))$. The complexity of each query of $G(\alpha_k,\beta_k)$ is $\text{polylog}(mn)$, thus the complexity of Step 3) is $O(\max_{ij}L_{ij}\text{polylog}(mn))=O(\frac{h}{\varepsilon_0}\log_2(1/\delta')\text{polylog}(mn))$.

The complexity of Step 6 can be neglected.

To ensure that the error of $|\psi\rangle$ is within ε, we could just let $\delta'=O(\varepsilon)$. As a conclusion, the complexity of the algorithm is $O(\frac{h}{\varepsilon_0}\text{polylog}(mn/\varepsilon))$.

Appendix B: A brief complexity analysis of the first two sub-algorithms of VQNPE

For the first sub-algorithm of VQNPE, *i.e.*, the algorithm to find the k-nearest neighbors, the authors used swap test circuits to obtain the square of the inner product of each pair of data points, which was regarded as the distance between data points by the following steps. Here we should mention that the authors implicitly assumed that $\|\boldsymbol{x}_i\|=1$, since they defined $|\boldsymbol{x}_i\rangle=\sum_j x_{ij}|j\rangle$. The complexity of this step should be $O(\frac{m^2}{\varepsilon^2}\log_2 n)$, where ε is the error of the square of inner products. Here we also point out that this method can not obtain the distance of the vectors such as $(1,0)^{\mathrm{T}}$ and $(-1,0)^{\mathrm{T}}$, thus we used a different method to get the distance without this problem.

Let $\boldsymbol{x}_j^{(i)}, j\in\{0,1,2,\ldots,k^{(i)}-1\}$ denotes the $k^{(i)}$ nearest neighbors of $\boldsymbol{x}_i$, $\mathcal{A}_i=(\boldsymbol{x}_i-\boldsymbol{x}_0^{(i)},\boldsymbol{x}_i-\boldsymbol{x}_1^{(i)},\ldots,\boldsymbol{x}_i-\boldsymbol{x}_{k^{(i)}-1}^{(i)})$, $C_1^{(i)}=\mathcal{A}_i^{\dagger}\mathcal{A}_i$. Then $C_1^{(i)}$ is a $k^{(i)}\times k^{(i)}$ matrix which is actually the matrix left by deleting the zero rows and columns of $C^{(i)}$, *i.e.* deleting j rows and j columns for $\boldsymbol{x}_j\notin Q_i$.

For the algorithm to obtain $|\boldsymbol{w}_i\rangle$, the authors assumed that there is an oracle to access the element of $\mathcal{A}_i$ for $i=0,1,\ldots,m-1$, that is,

$$|j\rangle|l\rangle|0\rangle\rightarrow|j\rangle|l\rangle|\mathcal{A}_{jl}^{i}\rangle=|j\rangle|l\rangle|x_{ij}-x_{lj}^{(i)}\rangle, \tag{B1}$$

where $\mathcal{A}_{jl}^{i}$ is the j-th row l-th column element of $\mathcal{A}_i$, x_{ij} is the j-th element of $\boldsymbol{x}_i$ and $x_{lj}^{(i)}$ is the j-th element of the lth nearest neighbor of $\boldsymbol{x}_i$.

With the oracle mentioned above, according to quantum singular value decomposition (QSVD),[57] since $\mathcal{A}_i$ is an $n\times k$ matrix, one could simulate $\mathrm{e}^{\mathrm{i}\frac{\hat{\mathcal{A}}_i}{n+k}t}$ with complexity $O(\frac{t^2}{\varepsilon}\|\hat{\mathcal{A}}_i\|_F)$, where

$$\hat{\mathcal{A}}_i=\begin{pmatrix}\mathbf{0} & \mathcal{A}_i\\ \mathcal{A}_i^{\dagger} & \mathbf{0}\end{pmatrix}. \tag{B2}$$

Let $\hat{\mathcal{A}}_i=\sum_{j=1}^{k}\sigma_{j\pm}|\psi_{j\pm}\rangle\langle\psi_{j\pm}|$, similar to Eq. (36). According to Ref. [57], the necessary condition for this algorithm to be efficient is that $\sigma_{j\pm}=\Theta(n+k)$ for $j\in\{1,2,\ldots,k\}$.

Let λ_j, $j\in\{1,2,\ldots,n+k\}$ denote all of the eigenvalue of $\hat{\mathcal{A}}_i$, it is obvious that $\sigma_{j\pm}$ is included by the set of λ_j. According to the Gershgorin circle theorem, for $j\leq n$,

$$|\lambda_j|\leq\sum_{l=1}^{k}|\mathcal{A}_{jl}|\leq\sum_{l=1}^{k^{(i)}}|x_{ij}-x_{lj}^{(i)}|. \tag{B3}$$

since $\|\boldsymbol{x}_i\| = 1$, we have $|x_{ij}| \le 1$, thus $|\lambda_j| \le \sum_{l=1}^{k} 2 = 2k$. Similarly, for $j > n$,

$$|\lambda_j| \le \sum_{l=1}^{n} |\mathcal{A}_{(j-n)l}| = \sum_{l} |x_{il} - x_{jl}^{(i)}| = \|\boldsymbol{x}_i - \boldsymbol{x}_j^{(i)}\|_1.$$

Since $\|\boldsymbol{x}_i\| = 1$, we have $\|\boldsymbol{x}_i - \boldsymbol{x}_j^{(i)}\| = \sum_l (x_{il} - x_{jl}^{(i)})^2 \le 2$, where $\boldsymbol{x}_j^{(i)}$ is the jth nearest neighbor of $\boldsymbol{x}_i$. According to the inequality $\|\boldsymbol{x}\|_1 \le \sqrt{n}\|\boldsymbol{x}\|$, we have $|\lambda_j| \le 2\sqrt{n}$. Thus we have $|\lambda_j| \le 2\sqrt{n}$ for all $j \in \{1, 2, \ldots, n+k\}$, which means that $\sigma_{i\pm} = O(\sqrt{n}) \neq \Theta(n+k)$.

Since $\sigma_{i\pm} \neq \Theta(n+k)$, the algorithm is not efficient, thus the complexity is of $\Omega(\mathrm{poly}(n))$.

References

[1] Shor P W 1994 *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, November 20–22, 1994, Washington, USA, p. 124

[2] Grover L K 1996 *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, May 22–24, 1996, Philadelphia, USA, p. 212

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

[4] Wan L C, Yu C H, Pan S J, Gao F, Wen Q Y and Qin S J 2018 *Phys. Rev. A* **97** 062322

[5] Zhao L M, Zhao Z K, Rebentrost P and Fitzsimons J 2021 *Quantum Machine Intelligence* **3** 21

[6] Lloyd S, Mohseni M and Rebentrost P 2013 arXiv: 1307.0411v2 [quant-ph]

[7] Rebentrost P, Mohseni M and Lloyd S 2014 *Phys. Rev. Lett.* **113** 130503

[8] Cong I and Duan L 2016 *New J. Phys.* **18** 073011

[9] Ye Z K, Li L Z, Situ H Z and Wang Y Y 2020 *Sci. China Inform. Sci.* **63** 189501

[10] Wiebe N, Braun D and Lloyd S 2012 *Phys. Rev. Lett.* **109** 050505

[11] Schuld M, Sinayskiy I and Petruccione F 2016 *Phys. Rev. A* **94** 022342

[12] Wang G 2017 *Phys. Rev. A* **96** 012335

[13] Yu C H, Gao F and Wen Q Y 2021 *IEEE T. Knowl. Data En.* **33** 858

[14] Yu C H, Gao F, Liu C, Huynh D, Reynolds M and Wang J 2019 *Phys. Rev. A* **99** 022301

[15] Yu C H, Gao F, Wang Q L and Wen Q Y 2016 *Phys. Rev. A* **94** 042311

[16] Liu N and Rebentrost P 2018 *Phys. Rev. A* **97** 042315

[17] Bishop C M 2006 *Pattern Recognition and Machine Learning (Information Science and Statistics)*, (1st edn.), (New York: Springer) pp. 24–28

[18] Hotelling H 1936 *Biometrika* **28** 321

[19] Fisher R A 1936 *Ann. Eugenics* **7** 179

[20] He X, Cai D, Yan S and Zhang H J 2005 *Tenth IEEE International Conference on Computer Vision*, Octobor 17–21, 2005, Beijing, China, p. 1208

[21] Roweis S T and Saul L K 2000 *Science* **290** 2323

[22] Lloyd S, Mohseni M and Rebentrost P 2014 *Nat. Phys.* **10** 631

[23] Yu C H, Gao F, Lin S and Wang J 2019 *Quantum Inf. Process.* **18** 249

[24] Duan B, Yuan J, Xu J and Li D 2019 *Phys. Rev. A* **99** 032311

[25] Pan S J, Wan L C, Liu H L, Wang Q L, Qin S J, Wen Q Y and Gao F 2020 *Phys. Rev. A* **102** 052402

[26] Li Y, Zhou R G, Xu R, Hu W and Fan P 2020 *Quantum Sci. Technol.* **6** 014001

[27] Meng F X, Yu X T, Xiang R Q and Zhang Z C 2019 *IEEE Access* **7** 4825

[28] Pan S J, Gao F, Wan L C, Qin S J and Wen Q Y 2021 *Journal of Computer Research and Development* **58** 1835 (in Chinese)

[29] Liang J M, Shen S Q, Li M and Li L 2020 *Phys. Rev. A* **101** 032323

[30] Cerezo M, Arrasmith A, Babbush R, Benjamin S C, Endo S, Fujii K, McClean J R, Mitarai K, Yuan X, Cincio L and Coles P J 2021 *Nat. Rev. Phys.* **3** 625

[31] Liu H L, Wu Y S, Wan L C, Pan S J, Qin S J, Gao F and Wen Q Y 2021 *Phys. Rev. A* **104** 022418

[32] Kerenidis I and Prakash A 2017 *8th Innovations in Theoretical Computer Science Conference*, January 9–11, 2017, Berkeley, USA, p. 49

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

[34] Chen J and Ma Z 2011 *International Journal of Pattern Recognition and Artificial Intelligence* **25** 985

[35] Cai D, He X and Han J 2007 *2007 IEEE 11th International Conference on Computer Vision*, October 14–21, 2007, Rio de Janeiro, Brazil, p. 1

[36] Cai D, He X and Han J 2007 *Proceedings of the 15th ACM International Conference on Multimedia*, September 24–29, Augsburg, Germany, p. 403

[37] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M and Duchesnay E 2011 *J. Mach. Learn. Res.* **12** 2825

[38] Brassard G, Hoyer P, Mosca M and Tapp A 2002 *Contemporary Mathematics* **305** 53

[39] Kerenidis I, Landman J, Luongo A and Prakash A 2019 *33rd Conference on Neural Information Processing Systems*, December, 8–14, 2019, Vancouver, Canada, p. 4134

[40] Vedral V, Barenco A and Ekert A 1996 *Phys. Rev. A* **54** 147

[41] Zhou S, Loke T, Izaac J and Wang J 2017 *Quantum Inf. Process.* **16** 82

[42] Cao Y, Papageorgiou A, Petras I, Traub J and Kais S 2013 *New J. Phys.* **15** 013021

[43] Erdös P 1961 *Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei* **6** 215

[44] Kerenidis I and Landman J 2021 *Phys. Rev. A* **103** 042415

[45] Kerenidis I and Prakash A 2020 *ACM Transactions on Quantum Computing* **1** 5:1

[46] Yin Q, Xiang G Y, Li C F and Guo G C 2018 *Chin. Phys. Lett.* **35** 070302

[47] Gilyén A, Su Y, Low G H and Wiebe N 2019 *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, June 23–26, Arizona, USA, p. 193

[48] Low G H and Chuang I L 2019 *Quantum* **3** 163

[49] Chakraborty S, Gilyen A and Jeffery S 2019 *46th International Col loquium on Automata, Languages and Programming*, July 9–12, 2019, Patras, Greece, p. 33:1

[50] Ghojogh B, Ghodsi A, Karray F and Crowley M 2020 arXiv: 2011.10925v1 [stat.ML]

[51] Durr C and Hoyer P 1996 arXiv: 9607014v2 [quant-ph]

[52] Li S, Liang J, Shen S and Li M 2021 *Commun. Theor. Phys.* **73** 105102

[53] Tang E 2019 *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, June 23–26, Arizona, USA, p. 217

[54] Chia N H, Gilyén A, Li T Y, Lin H H, Tang E and Wang C H 2020 *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, June 22–26, New York, USA, p. 387

[55] Chen Y L and De Wolf R 2021 arXiv: 2110.13086 [quant-ph]

[56] Yoder T J, Low G H and Chuang I L 2014 *Phys. Rev. Lett.* **113** 210501

[57] Rebentrost P, Steffens A, Marvian I and Lloyd S 2018 *Phys. Rev. A* **97** 012327