Quantum locally linear embedding for nonlinear dimensionality reduction

Xi He · Li Sun · Chufan Lyu · Xiaoting Wang

Received: date / Accepted: date

Abstract Reducing the dimension of nonlinear data is crucial in data processing and visualization. The locally linear embedding algorithm (LLE) is specifically a representative nonlinear dimensionality reduction method with well maintaining the original manifold structure. In this paper, we present two implementations of the quantum locally linear embedding algorithm (qLLE) to perform the nonlinear dimensionality reduction on quantum devices. One implementation, the linear-algebra-based qLLE algorithm utilizes quantum linear algebra subroutines to reduce the dimension of the given data. The other implementation, the variational qLLE algorithm utilizes a variational quantum-classical hybrid procedure to acquire the low-dimensional data. The classical LLE algorithm requires polynomial time complexity of $N$, where $N$ is the global number of the original high-dimensional data. In data preprocessing, we invoke the quantum $k$-nearest neighbors algorithm ($k$-NN) to find out the $k$ nearest neighbors of the given data with quadratic speedup. For the main part of the qLLE algorithm, compared with the corresponding classical algorithm, the linear-algebra-based qLLE algorithm proposed in this paper achieves an exponential speedup in $O(\text{poly}(\log N))$. The variational qLLE algorithm can be implemented on the near term quantum devices. In addition, the corresponding quantum circuits of the qLLE algorithm are also presented.

Keywords Locally linear embedding · Quantum dimensionality reduction · Quantum machine learning · Quantum computation

We thank Zizhu Wang for helpful discussions and suggestions.

Xi He, Li Sun, Chufan Lyu and Xiaoting Wang
Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 610054, China
E-mail: xihhe@std.uestc.edu.cn
1 Introduction

The dimensionality reduction in the field of machine learning refers to using some methods to map the data points in the original high-dimensional space into some low-dimensional space. The reason why we reduce the data dimension is that the original high-dimensional data contains redundant information, which sometimes causes errors in practical applications. By reducing the data dimension, we hope to reduce the error caused by the noise information and to improve the accuracy of identification or other applications. In addition, we also want to find the intrinsic structure of the data through the dimensionality reduction in some cases. Among the many dimensionality reduction algorithms, one of the most widely used algorithms is the principal component analysis (PCA). PCA is a linear dimensionality reduction algorithm embedding the given data into a linear low-dimensional subspace [1, 2]. In addition to the PCA algorithm, the linear discriminant analysis algorithm (LDA) [3] and the A-optimal projection algorithm (AOP) [4] both show outstanding performance in reducing the data dimension.

However, all the algorithms mentioned above actually belong to linear dimensionality reduction techniques. It means that they are not so efficient in dealing with nonlinear data to some extent. Thus, some nonlinear dimensionality reduction algorithms are proposed to give special treatment to nonlinear high-dimensional data. The locally linear embedding algorithm (LLE) which was proposed in 2000 by Sam T.Roweis and Lawrence K.Saul is typically one of the most representative nonlinear dimensionality reduction algorithms [5]. It can preserve the original topological structure of the data set during the process of dimensionality reduction. At present, LLE has been widely applied in all kinds of fields such as data visualization, pattern recognition and so on.

Although classical algorithms can accomplish the machine learning tasks effectively, quantum computing techniques can be applied to the realm of machine learning resulting in exponential speedup. Based on the quantum phase estimation algorithm [6], quantum algorithm for linear systems of equations [7], Grover’s algorithm [8] and so on, all kinds of quantum machine learning algorithms have been proposed. Representatively, quantum support vector machine [9], quantum data fitting [10] and quantum linear regression [11] were proposed respectively to deal with problems of pattern classification and prediction. Quantum deep learning algorithms such as quantum Boltzmann machine [12], quantum generative adversarial learning [13, 14] and so on were also put forward to exhibit their capabilities in quantum physics. In addition, there are some variational quantum-classical hybrid algorithms presented for machine learning recently [15, 16, 17]. In general, quantum machine learning has developed into a vibrant interdisciplinary field.

In the field of quantum dimensionality reduction, there are also some outstanding techniques. The quantum principal component analysis algorithm (qPCA) was proposed with exponential speedup compared with the classical PCA algorithm [18]. Afterwards, the quantum linear discriminant analysis (qLDA) was designed for dimensionality reduction and classification [19]. Recently, the quantum A-optimal projection algorithm (QAOP) presented in [20] performs superior regression performance. Different from all the above-mentioned algorithms, in this paper, we present two
implementations of the quantum locally linear embedding algorithm (qLLE) for nonlinear dimensionality reduction. Compared with the classical LLE algorithm, we can invoke the quantum k-NN algorithm to find out the k nearest neighbors of all the given data with quadratic speedup in the data preprocessing stage [21]. In the main part of the qLLE algorithm, the linear-algebra-based qLLE algorithm can be implemented in $O(\text{poly}(\log N))$, which achieves exponential speedup compared with the classical LLE algorithm in time $O(\text{poly}(N))$ where $N$ is the number of the original high-dimensional data points. The variational qLLE algorithm can be performed on the near term quantum devices.

In our work, the contributions are mainly reflected in two aspects. One contribution, two qLLE implementations are presented for nonlinear dimensionality reduction. The linear-algebra-based qLLE can be performed on a universal quantum computer with logarithmic resources. The variational qLLE can be implemented on near term quantum devices without the requirement of fully coherent evolution. The other contribution, we design the corresponding quantum circuits for the qLLE algorithm making the execution of our theoretical analysis realizable.

In section 2, we briefly overview the classical LLE algorithm. Then, the qLLE algorithm is presented in detail. In addition to the theoretical analysis, we also provide the detailed corresponding quantum circuits. To analyze the performance of the algorithms, we discuss the algorithmic complexity of classical and quantum LLE algorithms in section 4. Finally, we make a conclusion in section 5.

2 Classical locally linear embedding

In this section, the classical LLE algorithm is described in detail [5]. The LLE algorithm aims to reduce the dimension of the high-dimensional data with embedding them from the original high-dimensional space to some low-dimensional space. During the procedure of dimensionality reduction, the linear relationships between the data points and their corresponding $k$ nearest neighbors remain unchanged. The schematic diagram of the classical LLE algorithm is presented in Fig. 1.

The overall procedure of LLE can be summarized as follows:

(1) Find out the $k$ nearest neighbors of each high-dimensional data point $x_i$ with $i = 1, 2, \ldots, N$.

(2) Construct the local reconstruction weight matrix $W$ with the $k$ nearest neighbors of $x_i$.

(3) Compute the low-dimensional data $Y$ with the reconstruction weight matrix $W$.

Herein, we review the classical LLE algorithm in detail. Suppose we have the input data set $X = \{x_i \in \mathbb{R}^D : 1 \leq i \leq N\}$. LLE is a local dimensionality reduction algorithm which mainly utilizes the local linearity among the data points to approximate their global property. Thus, in the first place, LLE attempts to find out the $k$ nearest neighbors of each data point with the $k$-NN algorithm to subsequently construct the reconstruction weight matrix $W$. As a matter of fact, $W$ is the intermediary that connects the entire dimensionality reduction procedure. Having acquired the $k$
The high-dimensional data are distributed in a manifold $S_1$ and each data point $x_i$ is assumed to be represented by the linear combination of its $k$ nearest neighbors $x_j^{(i)}$. The LLE algorithm aims to find the low-dimensional data $y_i$ embedded in $S_2$ with keeping this linear relationship, which means that the weight matrix $W$ is invariable during the dimensionality reduction.

For the nearest neighbors of all the data points, we can then set the cost function

$$
\min_W \Phi(W) = \sum_{i=1}^N \left\| x_i - \sum_{j=1}^k W_{ij} x_j^{(i)} \right\|_F^2
$$

$$
s.t. \sum_{j=1}^k W_{ij} = 1
$$

where $x_j^{(i)}$ represents the $j$th nearest neighbor of $x_i$, and $W_{ij}$ denotes the weight coefficient of $x_j^{(i)}$.

Equivalently, the matrix form of the cost function $\Phi(W)$ is

$$
\min_W \Phi(W) = \sum_{i=1}^N W_i^T (\Delta X_i)(\Delta X_i)^T W_i = \sum_{i=1}^N W_i^T C_i W_i
$$

$$
s.t. W_i^T 1_N = 1
$$
where \( W_i = (W_{i1}, W_{i2}, \ldots, W_{ik}, 0, \ldots, 0)^T \) is a \( N \)-dimensional vector; \( \Delta X_i = [(x_i - x^{(i)}_1), \ldots, (x_i - x^{(i)}_k), 0, \ldots, 0] \) is a \( D \times N \) matrix; \( C_i = (\Delta X_i)^T(\Delta X_i) \) and \( 1_N = (1, 1, \ldots, 1)^T \).

Applying the method of Lagrangian multiplier on Eq. (2), we can obtain

\[
W_i = \frac{C_i^{-1}1_N}{1_N^T C_i^{-1}1_N},
\]

and the specific derivation of \( W_i \) is presented in Appendix A.

In practice, we can efficiently get the \( W_i \) with solving the linear system of equations \( C_i W_i = 1_N \), and then rescaling the weights for normalization. Hence, the reconstruction weight matrix \( W = (W_1, W_2, \ldots, W_i, \ldots, W_N)_{N \times N} \) can be subsequently constructed.

Assuming the low-dimensional data set after dimensionality reduction is \( Y = \{y_i \in \mathbb{R}^d : 1 \leq i \leq N\} \), we want to keep the linear relationship during the process of dimensionality reduction. It is equivalent to minimizing the cost function

\[
\min_Y \Phi(Y) = \sum_{i=1}^{N} \|y_i - \sum_{j=1}^{k} W_{ij} y^{(i)}_j \|_F^2,
\]

\[
\text{s.t. } \sum_{i=1}^{N} y_i = 0;
\]

\[
\frac{1}{N} \sum_{i=1}^{N} y_i y_i^T = I_d,
\]

where \( y^{(i)}_j \) represents the \( j \)th nearest neighbor of \( y_i \); \( W_{ij} \) denotes the weight coefficient of \( y^{(i)}_j \) and is exactly the same as the weight coefficient in Eq. (1).

Similarly, Eq. (4) can be transformed to its matrix form

\[
\min_Y \Phi(Y) = \|Y - YW\|_F^2,
\]

\[
\text{s.t. } Y 1_N = 0;
\]

\[
\frac{1}{N} Y Y^T = I_d,
\]

where \( 1_N = (1, 1, \ldots, 1)^T_N \) is a \( N \)-dimensional vector with all the elements equaling 1.

Having set \( \Phi(Y) \) properly, we can subsequently transform Eq. (5) into

\[
\min \Phi(Y) = \text{tr}(YMY^T),
\]

where the target matrix \( M = (I_N - W)(I_N - W^T) \). The derivation with Lagrangian multiplier is presented in Appendix B.

Therefore, the problem of minimizing \( \Phi(Y) \) can be transformed into solving the 2nd to the \((d + 1)\)th smallest eigenvectors of \( M \)(the first eigenvalue of \( M \) is 0, and the corresponding eigenvector is \( 1_N \), so it does not reflect the data characteristics). Finally, we get the reduced dimension data set \( Y = (y_1, y_2, \ldots, y_N) = \).
$(u_2, u_3, ..., u_{d+1})^T$, where $u_2, u_3, ..., u_{d+1}$ stand for the 2nd to the $(d+1)$th eigenvectors of $M$ which are corresponding to the relative smallest eigenvalues in ascending order.

### 3 Quantum locally linear embedding

In this section, we demonstrate the specific implementation procedure of the quantum locally linear embedding algorithm (qLLE). In the first place, we encode the data points to quantum states and find the $k$ nearest neighbors of them with the quantum $k$-NN algorithm. After the data preprocessing, the reconstruction weight matrix $W$ will be constructed with quantum linear algebra subroutines. Having obtained $W$, we can subsequently prepare the target matrix $M$ utilizing quantum matrix multiplication operations. At last, we proposed two implementations of the qLLE algorithm: one method is based on quantum linear algebra subroutines; the other method makes use of the variational quantum computation. In addition to the theoretical analysis, we also provide the corresponding quantum circuits.

#### 3.1 Data preprocessing

To present the qLLE algorithm, the input data vector $x_i = (x_{1i}, x_{2i}, ..., x_{Di})^T$ can be encoded as a $q$-qubit quantum state $|x_i⟩$ where $q = \log D$. Assume that the quantum states $|x_j^{(i)}⟩$ for $j = 1, ..., k$ represent the $k$ nearest neighbors of $|x_i⟩$. With invoking the quantum $k$-NN algorithm, we can find out the $k$ nearest neighbors $\{|x_j^{(i)}⟩: 1 \leq j \leq k\}$ of $|x_i⟩$ for $i = 1, 2, ..., N$. Compared with the classical $k$-NN algorithm which should construct the corresponding data structure index in $O(N \log N)$ and search the $k$ nearest neighbors in $O(k \log N)$, the quantum $k$-NN algorithm can be implemented with quadratic speedup in $O(R \sqrt{kN})$ where $R$ is the times of Oracle execution \[21\].

#### 3.2 Construction of the weight matrix $W$

With the procedure of data preprocessing, all the original quantum states and their corresponding $k$ nearest neighbors can be obtained. Subsequently, we present how to construct the weight matrix $W$.

In the first place, we prepare the quantum states $|x_i - x_j^{(i)}⟩$ with the data points $x_i$ and the corresponding nearest neighbors $x_j^{(i)}$ where $i = 1, 2, ..., N$ and $j = 1, 2, ..., k$. It is note that the elements which are not corresponding to the $k$ nearest neighbors are not under consideration according to section \[2\]. With knowing all the elements of $x_i$, the corresponding $q$-qubit quantum state $|x_i⟩ = |a_qa_{q-1}...a_1⟩$. We apply the quantum Fourier transform (QFT) on $|x_i⟩$ resulting in

$$
|a_qa_{q-1}...a_1⟩ \xrightarrow{\text{QFT}} |\phi_q(a)⟩ \otimes |\phi_{q-1}(a)⟩ \otimes \cdots \otimes |\phi_2(a)⟩ \otimes \cdots \otimes |\phi_1(a)⟩, \quad (7)
$$
Fig. 2 The quantum circuit of the quantum subtractor computing $|x_i - x_j(i)\rangle$

where $|\phi_p(a)\rangle = \frac{1}{\sqrt{2^q}}(|0\rangle + e^{2\pi i a_p \cdot a_{p-1} \ldots a_1}|1\rangle)$ for $p = 1, 2, \ldots, q$ \[22\].

Similarly, $x_j(i)$ can be encoded as $|x_j(i)\rangle = |b_q b_{q-1} \ldots b_1\rangle$. Subsequently, we perform the controlled rotation operation $U_P = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes R_P$ as shown in Fig. 2 to prepare the quantum state $|\phi_q(a - b)\rangle \otimes |\phi_{q-1}(a - b)\rangle \otimes \cdots \otimes |\phi_1(a - b)\rangle$, where $R_P = \begin{bmatrix} 1 & 0 \\ 0 & e^{-2\pi i/2^p} \end{bmatrix}$ with $p = 1, 2, \ldots, q$. It is noted that the controlled rotation operation of the quantum subtractor we adopt can be performed in parallel with $q \log q$ operations \[23, 24, 25\]. After performing the inverse QFT on $|\phi_q(a - b)\rangle \otimes |\phi_{q-1}(a - b)\rangle \otimes \cdots \otimes |\phi_1(a - b)\rangle$, the quantum state $|x_i - x_j(i)\rangle = |a_q - b_q\rangle \otimes |a_{q-1} - b_{q-1}\rangle \otimes \cdots \otimes |a_1 - b_1\rangle$ can be obtained. With iteratively invoking the quantum subtractor, we can finally acquire all the quantum states $|x_i - x_j(i)\rangle$ with $j = 1, 2, \ldots, k$. The quantum circuit of the quantum subtractor is presented in Fig. 2.

In addition to solving the quantum states $|x_i - x_j(i)\rangle$, we also need to compute the norms $|x_i - x_j(i)|$. It is obvious that

$$|x_i - x_j(i)|^2 = |x_i|^2 + |x_j(i)|^2 - 2\text{Re}\langle x_i, x_j(i)\rangle. \quad (8)$$

With the given data and the procedure of data preprocessing, the norms $|x_i|$ and $|x_j(i)|$ are trivial. As to the third term of Eq. (8), we firstly prepare the initial state

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle |x_i\rangle + |1\rangle |x_j(i)\rangle). \quad (9)$$

Then, the Hadamard gate is performed on the ancilla to obtain the state

$$|\psi_1\rangle = \frac{1}{2} \left[ |0\rangle (|x_i\rangle + |x_j(i)\rangle) + |1\rangle (|x_i\rangle - |x_j(i)\rangle) \right]. \quad (10)$$
The quantum circuit of computing the inner product \( \langle x_i | x_j^{(i)} \rangle \)

We measure the ancilla resulting in the probability of \(|0\rangle \)

\[
P_1(0) = \frac{1}{4} (\langle x_i | + \langle x_j^{(i)} |)(| x_i | + | x_j^{(i)} |))
= \frac{1}{4} (2 + \langle x_i | x_j^{(i)} | + \langle x_j^{(i)} | x_i |)
= \frac{1}{2} + \frac{1}{2} \text{Re} \langle x_i | x_j^{(i)} |.
\]
(11)

Thus, the third term of Eq. (8)

\[
2 \text{Re} \langle x_i | x_j^{(i)} | = 2 |x_i| | x_j^{(i)} | (x_i | x_j^{(i)} |)
= (4P_1(0) - 2) |x_i| | x_j^{(i)} |.
\]
(12)

In practice, we achieve this procedure with the swap test circuit as shown in Fig. 3[26]. It is noted that the input state is \(|0\rangle | x_i \rangle | x_j^{(i)} \rangle \) and we trace out the third register before the measurement to acquire the state \(|\psi_1\rangle \). Finally, we can compute the inner product with the probability of measuring \(|0\rangle \) on the ancilla qubit [27].

Having access to \(|x_i - x_j^{(i)} \rangle \) and \(|x_i - x_j^{(i)} \rangle \), we can utilize the quantum random access memory (qRAM) to construct the state

\[
|\psi_{\Delta X_i} \rangle = \frac{1}{\sqrt{\sum_{j=1}^k \sum_{m=1}^D (x_{mi} - x_{mj}^{(i)})^2}} \sum_{j=1}^k \sum_{m=1}^D (x_{mi} - x_{mj}^{(i)}) |j\rangle |m\rangle
\]
(13)

where \(\Delta X_i = [(x_i - x_1^{(i)}) \ldots (x_i - x_k^{(i)})] \). Afterwards, we trace out the \(|m\rangle \) register resulting in the density operator

\[
\rho_{\tilde{C}_i} = \text{tr}_m \{|\psi_{\Delta X_i} \rangle \langle \psi_{\Delta X_i} | \}
= \frac{1}{\sum_{j=1}^k \sum_{m=1}^D (x_{mi} - x_{mj}^{(i)})^2} \sum_{j=1}^k \sum_{m=1}^D (x_{mi} - x_{mj}^{(i)}) (x_{mi} - x_{mj}^{(i)})^* |j\rangle \langle j'|
= \frac{\tilde{C}_i}{\text{tr}C_i},
\]
(14)

where \(\tilde{C}_i = \Delta X_i^T \Delta X_i [11]\).

In the last step, we attempt to construct the weight matrix \(W\) with repeatedly solving the linear equation \(C_i W_i = 1_N\) for \(i = 1, 2, \ldots, N\). Before applying the quantum
algorithm for linear systems of equations (HHL), we should firstly extend the matrix
\[ \tilde{C}_i \text{ to } C_i = \begin{bmatrix} \tilde{C}_i & 0 \\ 0 & 0 \end{bmatrix} \] which is obviously \( k \)-sparse. Finally, as a matter of fact, our
goal is to solve \( |W_i\rangle = \tilde{C}_i^{-1} |I_N\rangle \) with \( |W_i\rangle = (W_{i1}, W_{i2}, \ldots, W_{ik}, 0, \ldots, 0)^T \) and \( |I_N\rangle = \frac{1}{\sqrt{N}} (1, 1, \ldots, 1)^T \). We input the state \( |\psi_{init}\rangle = |0\rangle^{R_1} |0, \ldots, 0\rangle^C |I_N\rangle^B_1 \)
where \( n = \log N \), and perform the HHL algorithm on it to obtain \( |\tilde{W}_i\rangle \) [7]. Before
detailedly describing the procedure of solving \( |W_i\rangle \), we generalize the HHL algorithm at first.

As shown in Fig. 4 [28], \( \text{U}_{\text{HHL}} (M, f(\lambda)) = (I^R \otimes U_{\text{PE}}^\dagger (M)) (U_{\text{CR}}^\dagger (M, f(\lambda)) \otimes I^B) (I^R \otimes U_{\text{PE}} (M)) \) represents the unitary evolution of the HHL algorithm where the
unitary operation \( U_{\text{PE}} (M) = (F^\dagger \otimes I^B) \left( \sum_{\tau=0}^{T-1} |\tau\rangle \langle \tau|^C \otimes e^{i M t_\tau / T} \right) (H^H \otimes I^B) \)
represents the phase estimation algorithm performed with a specified sparse matrix \( M \) (It is note that \( F^\dagger \) stands for the inverse QFT) and \( U_{\text{CR}} (M, f(\lambda)) \) represents a conditional rotation operation which is specifically
\[
|0\rangle^R |\lambda\rangle^C \rightarrow \left( \sqrt{1 - \gamma^2 f(\lambda)^2} |0\rangle + \gamma f(\lambda) |1\rangle \right)^R \otimes |\lambda\rangle^C,
\]
where the register \( R \) is controlled by the register \( C \); \( \gamma \) is a constant; \( \lambda \) represents the
eigenvalue of \( M \) and \( f(\lambda) \) is a specified function about \( \lambda \) [29]. Thus, the overall
procedure of solving \( |W_i\rangle \) is presented as follows:

\[
|\psi_{init}\rangle \xrightarrow{U_{\text{PE}} (C_i)} |0\rangle^R \sum_{i=1}^{N} (u_{i1}|1_i\rangle/|\lambda_{1i}\rangle)^C_1 |u_{i1}\rangle^B_1
\]

\[
\xrightarrow{U_{\text{CR}} (C_i, \lambda^{-1})} \left( \sqrt{1 - \frac{\gamma^2}{\lambda_{1i}^2}} |0\rangle + \frac{\gamma}{\lambda_{1i}} |1\rangle \right)^R \sum_{i=1}^{N} (u_{i1}|1_i\rangle/|\lambda_{1i}\rangle)^C_1 |u_{i1}\rangle^B_1
\]

\[
\xrightarrow{\text{Uncompute}} \left( \sqrt{1 - \frac{\gamma^2}{\lambda_{1i}^2}} |0\rangle + \frac{\gamma}{\lambda_{1i}} |1\rangle \right)^R \sum_{i=1}^{N} (u_{i1}|1_i\rangle/|\lambda_{1i}\rangle)^B_1
\]

\[
\text{Measurement} \quad |W_i\rangle = \frac{1}{\sqrt{\sum_{i=1}^{N} \gamma_{1i}^2 / \lambda_{1i}^2}} \sum_{i=1}^{N} \frac{\gamma_{1i}}{\lambda_{1i}} (u_{i1}|1_i\rangle/|u_{i1}\rangle),
\]

Fig. 4 The quantum circuit of the HHL algorithm [28]
where $\gamma_i$ is a constant; $\lambda_i$ are the eigenvalues of $C_i$ and $u_i$ are the corresponding eigenvectors. With repeating this process for $i = 1, 2, \ldots N$, we can acquire all the columns of $W$. The quantum circuit of solving $W_i$ is depicted in Fig. 5.

### 3.3 Preparation of the target matrix $M$

Almost the same as the preparation of $|x_i - x_i^{(i)}\rangle$, we input the quantum states $|e_i\rangle$ and $|W_i\rangle$ into the quantum subtractor circuit resulting in $|e_i - W_i\rangle$ where $e_i$ is the $i$th column of the identity matrix $I_N$. We input the state $|0\rangle|e_i\rangle|W_i\rangle$ into the swap test circuit and trace out the $|W_i\rangle$ register. The probability of achieving $|0\rangle$ after measuring the ancilla is $P_2(0) = \frac{1}{2} + \frac{1}{2}\text{Re}\langle e_i|W_i\rangle$. Thus, the norm

$$|e_i - W_i| = \sqrt{2 - 2\text{Re}\langle e_i|W_i\rangle} = 2\sqrt{1 - P_2(0)}.$$  \hspace{1cm} (17)

Having acquired $|e_i - W_i\rangle$ and $|e_i - W_j\rangle$, we have quantum access to the matrix $I_N - W = \sum_{i=1}^{N} |e_i - W_i\rangle\langle e_i - W_i|$. In the next, we apply the $U_{\text{HHL}}(I_N - W, \lambda)$ on the quantum state $\rho_0 = \frac{1}{\sqrt{N}}\sum_{i=1}^{N} |i\rangle\langle i|$ to obtain the density operator $\rho_M$ which is proportional to the target matrix $M = (I_N - W)(I_N - W)^T$ according to section 2 [19]. Specifically, this procedure can be summarized as follows:

$$\rho_M = \frac{1}{\sqrt{\gamma_1^2 \gamma_2^2 \lambda_2^2 \lambda_2^2}} \sum_{i,j=1}^{N} \gamma_2 \gamma_2' \lambda_2 \lambda_2' \langle u_{2i}|\rho_0|u_{2j}\rangle \langle u_{2j}|\rho_0|u_{2i}\rangle,$$  \hspace{1cm} (18)
where the ancilla
\[
\begin{align*}
|\psi_{\text{anc}}\rangle &= \sqrt{1 - \gamma_2^2 \lambda_2^2} |0\rangle + \gamma_2 \lambda_2 i |1\rangle; \\
|\psi_{\text{anc}}\rangle &= \sqrt{1 - \gamma_2^2 \lambda_2^2} \sqrt{\lambda^2} |0\rangle + \gamma_2 \lambda_2^* i |1\rangle;
\end{align*}
\tag{19}
\]
\(\gamma_2, \gamma_2^*\) are constants; \(\lambda_2\) are the eigenvalues of \(I_N - W\) and \(u_2\) are the corresponding eigenvectors. In addition, the quantum circuit of this procedure is presented in Fig. 6.

It is noted that we can also compute the \(\rho_M\) utilizing the qRAM just like the preparation of \(\tilde{C}_i\). We can firstly construct the state
\[
|\psi_{I_N - W}\rangle = \frac{1}{\sum_{i=1}^{N} \sum_{m=1}^{N} |e_{mi} - W_{mi}|^2} \sum_{i=1}^{N} \sum_{m=1}^{N} (e_{mi} - W_{mi}) |i\rangle |m\rangle,
\tag{20}
\]
and trace out the \(|i\rangle\) register which is different from the Eq. (14) resulting in
\[
\rho_M = \text{tr}_i \{|\psi_{I_N - W}\rangle \langle \psi_{I_N - W}|\}
= \frac{1}{\sum_{i=1}^{N} \sum_{m=1}^{N} |e_{mi} - W_{mi}|^2} \sum_{i=1}^{N} \sum_{m,m'=1}^{N} (e_{mi} - W_{mi})(e_{mi} - W_{mi})^* |m\rangle \langle m'|
= \frac{M}{\text{tr}M^*},
\tag{21}
\]
ultimately, we obtain the density operator \(\rho_M\) according to Eq. (18) or Eq. (21). It is noted that our goal is to solve \(M\)'s second to \((d + 1)\)th smallest eigenvectors which are actually exactly the same as \(\rho_M\)'s corresponding eigenvectors. Thus, \(\rho_M\) has completely met our subsequent needs.

3.4 Linear-algebra-based qLLE

In this section, we propose an implementation of the qLLE algorithm utilizing quantum linear algebra subroutines.

Before applying the quantum linear algebra techniques, some modifications should be made. We define the matrix \(J = \xi I_N - \rho_M\) for some large \(\xi\). According to the analysis in section 3.3 it is obvious that solving the eigenvectors which are corresponding to \(M\)'s second to \((d + 1)\)th smallest eigenvalues is equivalent to finding the eigenvectors which are corresponding to \(J\)'s second to \((d + 1)\)th largest eigenvalues. Hence, our goal at present is to reveal the eigenvectors corresponding to the second to \((d + 1)\)th dominant eigenvalues of \(J\).
Because $J = \xi I_N - \rho_M$, the exponentiation of $J$

$$e^{-iJ\Delta t} = e^{-i\xi I_N \Delta t} e^{i\rho_M \Delta t} + O(\Delta t^2).$$

The matrix $\xi I_N$ is trivial. The procedure of applying $e^{i\rho_M \Delta t}$ which is equivalent to $e^{-i(-\rho_M)\Delta t}$ to any density operator $\sigma$ can be realized with utilizing the qPCA algorithm as follows

$$\sum_{l=0}^L |L\Delta t\rangle \langle L\Delta t| \otimes e^{-il(-\rho_M)\Delta t} \sigma e^{il(-\rho_M)\Delta t}$$

where $e^{-i(-\rho_M)\Delta t} \sigma e^{i(-\rho_M)\Delta t} \approx tr_1 \{e^{-iS\Delta t}(-\rho_M) \otimes \sigma e^{iS\Delta t}\} = \sigma - i\Delta t[-\rho_M, \sigma] + O(\Delta t^2)$ with $S = \sum_{i,j=1}^N |i\rangle \langle j| \otimes |j\rangle \langle i|$ being the swap operator; $tr_1$ is the partial trace over the first variable, and $L$ is the total evolution time with repeating this $\Delta t$ process $L$ times [18].

We can embed this density operator exponentiation process into the phase estimation algorithm resulting in $\sum_{i=1}^N \lambda_i |u_i\rangle \langle u_i| \otimes |\lambda_i\rangle \langle \lambda_i|$ where $u_i$ are the eigenvectors of $J$ and $\lambda_i$ are the corresponding eigenvalues. The second to $(d + 1)$th eigenvectors corresponding the dominant eigenvalues can be subsequently obtained with sampling from this state. Ultimately, the final $d$-dimensional data set $Y = \{y_i \in \mathbb{R}^d : 1 \leq i \leq N\} = (u_2, \ldots, u_{d+1})^T$.

3.5 Variational qLLE

In addition to the linear-algebra-based qLLE, we can alternatively implement the qLLE with the variational quantum computation. The variational qLLE algorithm is actually a variational quantum-classical hybrid algorithm designed for near term devices. The overall procedure of the variational qLLE algorithm is presented as follows.

(1) Prepare the ansatz states $|\psi(\lambda)\rangle$. As a matter of fact, the ansatz states $|\psi(\lambda)\rangle$ represents a set of parameterized circuits in practice where $\lambda$ represents a set of parameters. The ansatz states are prepared to introduce parameters to the cost function
Quantum locally linear embedding for nonlinear dimensionality reduction

Fig. 8 The schematic diagram of the variational qLLE algorithm

$E(\lambda_\theta)$. Specifically, we apply some parameterized quantum rotation operations on the ground states and entangle all the input states together resulting in the final ansatz states \[30\]. The specific quantum circuit of preparing the ansatz states is presented in Fig.7.

(2) Construct the cost function $E(\lambda_\theta)$. The cost function

$$E(\lambda_\theta) = \langle \psi(\lambda_\theta) | \rho_M | \psi(\lambda_\theta) \rangle + \sum_{i=1}^{\theta-1} \alpha_i |\langle \psi(\lambda_\theta) | \psi(\lambda_i) \rangle|^2,$$

where $\theta = 2, \ldots, d+1$; $\lambda_1 = 0$ and $\alpha_i$ are the corresponding coefficients \[16\]. The expectation value term $\langle \psi(\lambda_\theta) | \rho_M | \psi(\lambda_\theta) \rangle$ can be estimated with a one-step phase estimation circuit \[31\]. The overlap term $|\langle \psi(\lambda_\theta) | \psi(\lambda_i) \rangle|^2$ can be estimated with the swap test circuit \[26, 30\].

(3) Invoke classical operations. Having finished all the quantum parts, we invoke the classical adder to sum over all the expectation value term and the overlap terms resulting in the cost function $E(\lambda_\theta)$. Subsequently, $E(\lambda_\theta)$ can be minimized with a classical optimizer to obtain the eigenvalue $\lambda_\theta$.

(4) Iterate the steps (1) to (3) for $\theta = 2, \ldots, d+1$, we can obtain the second to $d+1$th smallest eigenvalues of $\rho_M$ and their corresponding eigenvectors. The schematic diagram of the variational qLLE algorithm is presented in Fig.8

4 Algorithmic Complexity of LLE

We usually compute the complexity of the entire process of the algorithm to evaluate the resource required to run it. The complexity of the LLE algorithm is mainly determined by the number of data points $N$, the original data dimension $D$, the target data dimension $d$, and the number of nearest neighbors $k$. In this section, we discuss
in detail the computational complexity of the classical LLE algorithm and the qLLE algorithm respectively.

The classical LLE algorithm mainly contains three steps. Firstly, we want to find the $k$ nearest neighbors of each input data point. Generally, the classical $k$-NN algorithm needs to construct the corresponding data structure index in $O(N \log N)$ and then to search the $k$ nearest neighbors in $O(k \log N)$ [32]. Then, the classical LLE subsequently attempts to solve $N$ set of linear equations in size of $k \times k$ requiring computational complexity in $O(Nk^3)$. Finally, the corresponding complexity of the eigenvalue analysis is in $O(dN^2)$. In summary, the overall complexity of the classical LLE algorithm is the sum of all these operations mentioned above [33].

In contrast with the classical LLE algorithm, we can obtain the computational complexity of the qLLE algorithm as follows. In data preprocessing, we invoke the quantum $k$-NN algorithm to find out the $k$ nearest neighbors of the original data in $O(R\sqrt{kN})$ where $R$ represents the Oracle execution times. Compared with the classical $k$-NN algorithm, this algorithm can achieve quadratic speedup [21]. As to the main part of the qLLE algorithm, we solve the weight matrix $W$ with performing the HHL algorithm $N$ times on $C_i$ for $i = 1, 2, \ldots, N$ in $O(\sum_{i=1}^{N} \kappa_i^2 \log N/\epsilon_1)$ where $\epsilon_1$ is the error parameter and $\kappa_i$ is the condition number of $C_i$ [7]. Subsequently, the target matrix $M$ can be prepared in $O(\kappa_0^3 \log N/\epsilon_2^3)$ where $\kappa_0$ is the condition number of the matrix $I_N - M$ and $\epsilon_2$ is the tolerance error [19]. It is worth mentioning that this quantum matrix multiplication operation for preparing $M$ achieves exponential speedup compared with the classical matrix multiplication in $O(N^3)$. In the last step, the $d$ principal components can be obtained with performing the qPCA algorithm on the target matrix $M$ in $O(d \log D)$ [18].

In summary, in addition to the data preprocessing where the quantum $k$-NN algorithm can quadratically reduce the resources in contrast to the classical $k$-NN algorithm, the linear-algebra-based qLLE algorithm can achieve exponential speedup compared with the classical LLE algorithm in each steps resulting in the overall exponential acceleration.

5 Conclusion

In this paper, we have presented a quantum version of the LLE algorithm which is a representative nonlinear dimensionality reduction algorithm. In data preprocessing, we invoke the quantum $k$-NN algorithm to find out the $k$ nearest neighbors of the original high-dimensional data with quadratic speedup. As to the main part of the algorithm, compared with the classical LLE algorithm, the qLLE algorithm can achieve exponential speedup. In addition to the theoretical analysis, the corresponding quantum circuits are also presented. Inspired by the VQE, we present an alternative in solving the eigenvalues and eigenvectors of the target matrix $M$ utilizing a variational quantum-classical hybrid algorithm. The variational qLLE algorithm can be implemented on the near term quantum devices and the schematic diagram is also presented. We attempt to explore the implementations as many as possible to inspire more works on the field of quantum dimensionality reduction.
A Derivation of \( W_i \)

According to the cost function Eq. (2), it is obvious that the Lagrange function
\[
\mathcal{L}_1(W_i, \mu) = W_i^T C_i W_i - \mu_1 (1 - W_i^T 1_N).
\]
where \( \mu_1 \) represents the Lagrange multiplier and \( i = 1, 2, \ldots, N \).

Let’s take the partial derivative of the Lagrange function \( \mathcal{L}_1 \) with respect to \( W_i \) and set it to zero
\[
\frac{\partial \mathcal{L}_1}{\partial W_i} = 2C_i W_i + \mu_1 1_N = 0
\]
resulting in
\[
W_i = -\frac{\mu_1}{2} C_i^{-1} 1_N.
\]
In addition, \( W_i^T 1_N = 1 \) is equivalent to \( 1_N^T W_i = 1 \). Thus, we have
\[
1_N^T (-\frac{\mu_1}{2} C_i^{-1} 1_N) = 1
\]
and then
\[
\mu_1 = -\frac{2}{1_N^T C_i^{-1} 1_N}
\]
Finally,
\[
W_i = \frac{C_i^{-1} 1_N}{1_N^T C_i^{-1} 1_N}
\]

B Derivation of \( Y \)

According to Eq. (5), we can get the Lagrange function
\[
\mathcal{L}_2(Y, \Lambda, \beta) = \frac{1}{2} \| Y - Y W \|_F^2 - \frac{1}{2} \text{tr}(A \left[ \frac{1}{N} Y Y^T - I_d \right]) - 1_N^T Y^T \beta,
\]
where \( A \) is a diagonal matrix whose diagonal elements are respectively the corresponding Lagrange multipliers and \( \beta = [\beta_1, \ldots, \beta_d]^T \) represents a \( d \)-dimensional vector whose elements are Lagrange multipliers.

We compute the partial derivative of the Lagrange function \( \mathcal{L}_2 \) with respect to \( Y \) and set it to zero
\[
\frac{\partial \mathcal{L}_2}{\partial Y} = Y(I_N - W)(I_N - W^T) - \frac{1}{N} \Lambda Y - \beta 1_N^T = 0
\]
Then, we multiply \( 1_N \) on both sides of Eq. (32) resulting in
\[
Y(I_N - W)(I_N - W^T) 1_N - \frac{1}{N} \Lambda Y 1_N - \beta 1_N^T 1_N = 0
\]
According to the constraint conditions of \( W \) and \( Y \), namely \( W_i^T 1_N = 1 \) and \( Y 1_N = 0 \), the first and the second term of Eq. (33) equal zero. Thus, \( \beta = 0 \) because of \( 1_N^T 1_N = N \). Therefore,
\[
Y M = \frac{1}{N} \Lambda Y,
\]
where \( M = (I_N - W)(I_N - W^T)^T \).

We diagonalize the matrix \( A = U D U^T \), with \( U \) is an orthogonal matrix and \( D = \text{diag}(d_1, d_2, \ldots, d_d) \) with \( d_1 \leq d_2 \leq \cdots \leq d_d \). We let \( Y = U^T Y \). Then Eq. (33) can be transformed to
\[
MY^T = \frac{1}{N} Y^T U DU^T
\]
which is equivalent to
\[ M\hat{Y} \hat{T} = \frac{1}{N} \hat{Y} \hat{T} D, \]  
(36)

where \( \hat{Y} = U^T Y \).

Eq. (35) shows that the columns of \( \hat{Y} \hat{T} \) (or the rows of \( \hat{Y} \')) are eigenvectors of \( M \). Because of \( N^{-1/2} \hat{Y} \hat{T} \hat{Y} \hat{T} = I_d \), the norm of each column of \( \hat{Y} \hat{T} \) equals \( N^{1/2} \). Hence, the columns of \( N^{-1/2} \hat{Y} \hat{T} \) are normalized eigenvectors of \( M \).

References

1. Pearson, K.: LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2(11), 559–572 (1901)
2. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology 24(6), 417 (1933)
3. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7(2), 179-188 (1936)
4. He, X., Zhang, C., Zhang, L., Li, X.: A-optimal projection for image representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(5), 1009-1015 (2015)
5. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323-2326 (2000)
6. Nielsen, M.A., Chuang, I.: Quantum Computation and Quantum Information (2002)
7. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. Physical Review Letters 103(15), 150502 (2009)
8. Grover, L.K.: A fast quantum mechanical algorithm for database search. arXiv preprint quant-ph/9605043 (1996)
9. Rebentrost, P., Mohseni, M., Lloyd, S.: Quantum support vector machine for big data classification. Physical Review Letters 113(13), 130503 (2014)
10. Wiebe, N., Braun, D., Lloyd, S.: Quantum algorithm for data fitting. Physical Review Letters 109(5), 050505 (2012)
11. Schuld, M., Smayskii, I., Petruccione, F.: Prediction by linear regression on a quantum computer. Physical Review A 94(2), 022342 (2016)
12. Amin, M.H., Andriyash, E., Rolfe, J., Kulchytskyy, B., Melko, R.: Quantum boltzmann machine. Physical Review X 8(2), 021050 (2018)
13. Lloyd, S., Weedbrook, C.: Quantum generative adversarial learning. Physical Review Letters 121(4), 040502 (2018)
14. Dallaire-Demers, P.-L., Killoran, N.: Quantum generative adversarial networks. Physical Review A 98(1), 012324 (2018)
15. Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.H., Zhou, X.Q., Love, P.J., Aspuru-Guzik, A., O’brien, J.L.: A variational eigenvalue solver on a photonic quantum processor. Nature Communications 5, 4213 (2014)
16. Higgott, O., Wang, D., Brierley, S.: Variational quantum computation of excited states. Quantum 3, 156 (2019)
17. LaRose, R., Tikku, A., O’Neel-Judy, É., Cincio, L., Coles, P.J.: Variational quantum state diagonalization. npj Quantum Information 5(1), 8 (2019)
18. Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum principal component analysis. Nature Physics 10(9), 631 (2014)
19. Cong, I, Duan, L.: Quantum discriminant analysis for dimensionality reduction and classification. New Journal of Physics 18(7), 073011 (2016)
20. Duan, B., Yuan, J., Xu, J., Li, D.: Quantum algorithm and quantum circuit for a-optimal projection: Dimensionality reduction. Physical Review A 99(3), 032311 (2019)
21. Dang, Y., Jiang, N., Hu, H., Yi, Z., Zhang, W.: Image classification based on quantum K-Nearest-Neighbor algorithm. Quantum Information Processing 17(9), 239 (2018)
22. Coppersmith, D.: An approximate fourier transform useful in quantum factoring. arXiv preprint quant-ph/0201067 (2002)
23. Barenco, A., Eckert, A., Suominen, K.A., Törmä, P.: Approximate quantum fourier transform and decoherence. Physical Review A 54(1), 139 (1996)
24. Zalka, C.: Fast versions of Shor’s quantum factoring algorithm. arXiv preprint quant-ph/9806084 (1998)
25. Draper, T. G.: Addition on a quantum computer. arXiv preprint quant-ph/0008033 (2000)
26. Buhrman, H., Cleve, R., Watrous, J., De Wolf, R.: Quantum fingerprinting. Physical Review Letters 87(16), 167902 (2001)
27. Schuld, M., Petruccione, F.: Supervised Learning with Quantum Computers, vol. 17. Springer (2018)
28. Dervovic, D., Herbster, M., Mountney, P., Severini, S., Usher, N., Wossnig, L.: Quantum linear systems algorithms: a primer. arXiv preprint arXiv:1802.08227 (2018)
29. Duan, B., Yuan, J., Liu, Y., Li, D.: Quantum algorithm for support matrix machines. Physical Review A 96(3), 032301 (2017)
30. Havlíček, V., Corcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., Gambetta, J. M.: Supervised learning with quantum-enhanced feature spaces. Nature 567(7747), 209 (2019)
31. Roggero, A., Baroni, A.: Short-depth circuits for efficient expectation value estimation. Roggero, Alessandro and Baroni, Alessandro arXiv preprint arXiv:1905.08383 (2019)
32. Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., Wu, A. Y.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of the ACM (JACM) 45(6), 891–923 (1998)
33. Saul, L. K., Roweis, S. T.: An introduction to locally linear embedding. unpublished. Available at: [http://www.cs.toronto.edu/~roweis/lle/publications.html](http://www.cs.toronto.edu/~roweis/lle/publications.html) (2000)