Quantum tensor singular value decomposition*

Xiaoqiang Wang, Lejia Gu, Heung-wing Lee and Guofeng Zhang
Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong
E-mail: guofeng.zhang@polyu.edu.hk

Keywords: tensor singular value decomposition, quantum singular value estimation, t-svd, t-product, quantum Fourier transform

Abstract
Tensors are increasingly ubiquitous in various areas of applied mathematics and computing, and tensor decompositions are of practical significance and benefit many applications in data completion, image processing, computer vision, collaborative filtering, etc. Recently, Kilmer and Martin propose a new tensor factorization strategy, tensor singular value decomposition (t-svd), which extends the matrix singular value decomposition to tensors. However, computing t-svd for high dimensional tensors costs much computation and thus cannot efficiently handle large scale datasets. Motivated by advantage of quantum computation, in this paper, we present a quantum algorithm of t-svd for third-order tensors and then extend it to order-\(p\) tensors. We prove that our quantum t-svd algorithm for a third-order \(N\) dimensional tensor runs in time \(O(N\text{polylog}(N))\) if we do not recover classical information from the quantum output state. Moreover, we apply our quantum t-svd algorithm to context-aware multidimensional recommendation systems, where we just need to extract partial classical information from the quantum output state, thus achieving low time complexity.

1. Introduction

A tensor (hypermatrix) refers to a multi-array of numbers. It has been applied in several areas including image deblurring, denoising, video recovery, data completion, tensor networks, multi-partite quantum systems, and machine learning [1–19], owing to the flexibility of tensors in representing data. Some of these applications utilize tensor decompositions including CANDECOMP/PARAFAC (CP) [20], tensor-train decomposition (TT) [21], TUCKER [22], higher-order singular value decomposition (HOSVD) [12, 23, 24], tensor singular value decomposition (t-svd) [1, 3, 19, 25], and etc.

During the last decade, plenty of research has been carried out on t-svd and its applications. The t-svd factorization strategy is first proposed by Kilmer and Martin [1] for third-order tensors, then it is extended to order-\(p\) tensors by Martin et al in 2013 [26]. The t-svd algorithm extends the matrix svd strategy to tensors while avoiding the loss of information inherent during unfolding tensors as operated in CP and TUCKER decompositions. The general idea of t-svd factorization is performing matrix svd in the Fourier domain, consequently it allows other matrix factorization techniques, e.g., QR decomposition, to be extended to tensors easily using the similar idea. Moreover, the t-svd is superior to Tucker or HOSVD in the sense that truncated t-svd gives an optimal approximation of a tensor measured by the Frobenius norm, while this best approximation cannot be obtained by truncating the full HOSVD or Tucker decomposition. Due to this optimality property, t-svd is shown to have better performance than HOSVD in facial recognition [27] and tensor completion [2, 28].

However, the complexity of calculating full t-svd for third-order \(N\) dimensional tensors is \(O(N^4)\), which is extremely high for large scale datasets. Hence many works have devoted to low-rank approximated t-svd representation which gives up the optimality property and has comparatively low complexity. In [29], Zhang et al propose a randomized t-svd method which can produce a factorization with similar properties to the t-svd, and the computational complexity is reduced to \(O(kN^3 + N^3 \log N)\), where \(k\) is the truncated term.

* This research is supported in part by Hong Kong Research Grant council (RGC) grants (No. 15208418, No. 15203619, No. 15506619) and Shenzhen Fundamental Research Fund, China under Grant No. JCYJ20190813165207290.
Considering the high cost of the existing classical algorithms of t-svd, we present a quantum version of t-svd for third-order tensors which reduces the complexity to $O(N \text{polylog}(N))$. To our best knowledge, the efficiency of this algorithm beats any known classical t-svd algorithms in the literature. In section 4, we extend the quantum t-svd algorithm to order-$p$ tensors.

An important step in a classical t-svd algorithm is to perform discrete Fourier transform (DFT) along the third mode of a tensor $A \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, obtaining $\hat{A}$ with computational complexity $O(N_3 \log N_3)$ for each tube $A(i, j, :)$, $i = 0, \cdots, N_1 - 1$; $j = 0, \cdots, N_2 - 1$. Thus, the complexity of performing the DFT on all tubes of the tensor $A$ is $O(N_1 N_2 N_3 \log N_3)$. In the quantum t-svd algorithm to be proposed, this procedure is accelerated by the quantum Fourier transform (QFT) whose complexity is only $O((\log N_3)^2)$. Moreover, due to quantum superposition, the QFT can be simultaneously performed on the third register of the state $|A\rangle$, which is equivalent to performing the DFT for all tubes of $A$ parallely, so the total complexity of this step is still $O((\log N_3)^2)$.

After performing the QFT, in order to further accelerate the second step in the classical t-svd algorithm which performs the matrix svd for every frontal slice of $A$, we apply a modified quantum singular value estimation (QSVE) algorithm originally proposed in [31] to the frontal slices $\hat{A}(; ; ;; i)$ parallely with complexity at most $O(N \text{polylog}(N))$ for $N$ dimensional tensors. Traditionally, the quantum singular value decomposition of non-sparse low-rank matrices involves exponentiating matrices and outputs the superposition state of singular values and their associated singular vectors. However, this Hamiltonian simulation method requires that the matrix to be exponentiated be low-rank, which is difficult to be satisfied in general. In our algorithm, we use the modified QSVE algorithm, where the matrix is unnecessarily low-rank, sparse or Hermitian.

The main contributions of this paper are listed as follows:

The original QSVE algorithm proposed in [31] has to be carefully modified to become a useful subroutine in our quantum tensor-svd algorithm. Specifically, the original QSVE, stated in lemma 3, requires the matrix $A$ be stored in the classical binary tree structure, then the singular values of $A$ can be estimated efficiently. Given a tensor $A \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, in algorithm 3, QSVE is performed on the matrices $\hat{A}^{(m)}$ which are frontal slices of $\hat{A}$, $m = 0, \cdots, N_3 - 1$. The difficulty lies in the fact that we cannot require all $\hat{A}^{(m)}$ be stored in the data structure since they are obtained after QFT. It is more reasonable to assume that the frontal slices of the original tensor $A$ is stored in the binary tree structure. Therefore, the main obstacle we need to overcome is how to estimate the singular values of $\hat{A}^{(m)}$ on the condition that every frontal slice of the original tensor $A^{(k)}$ is stored in the data structure. This problem is solved by theorem 2, and whose proof presents a detailed illustration of this process.

In section 5, we design a quantum tensor approximation algorithm based on algorithm 3, and present an application of this algorithm, namely the context-aware multidimensional recommendation systems. Our quantum tensor approximation algorithm suits the 3D recommendation systems model very well for mainly two reasons.

First, compared with other tensor decompositions, t-svd has been shown to be superior in capturing the spatial-shifting correlation [28], so it is suitable to model the 3D recommendation systems. Suppose the preference information of a user is encoded in a third-order tensor in which the three modes represent locations, point-of-interests, and time frames respectively, a user’s preference in a certain time is very likely to affect the recommendation for him/her at other time in general. The QFT in t-svd algorithm is used to bind a user’s preferences in different time together, so the recommendation using t-svd can better user experience by integrating the relations between different context.

Second, our quantum t-svd algorithm can achieve good result with low cost when applied to 3D recommendation systems. Actually, it is not necessary to reconstruct the entire tensor as in the classical 3D recommendation systems algorithms based on tensor factorizations, such as the truncated t-svd with complexity $O(kN^3 + N^3 \log N)$ [3] and the truncated HOSVD (T-HOSVD) with complexity $O(3kN^3)$ [32] for third-order $N$ dimensional tensors, where $k$ is the truncation rank. Our quantum 3D recommendation systems algorithm only samples high-value elements from the approximated tensor (corresponding to measuring the output state in computational basis a certain times), and this is exactly what we need for recommendation systems. Consequently, our quantum 3D recommendation systems algorithm achieves the complexity $O(N \text{polylog}N)$ if the preference tensor has several dominating (namely, high-value) elements.

The rest of this paper is organized as follows. A standard classical t-svd algorithm and several related concepts are introduced in section 2.2; section 2.3 summarizes the quantum singular value estimation algorithm proposed in [31]. Section 3 presents our main algorithm, quantum t-svd, and its complexity analysis. We extend the quantum t-svd algorithm to order-$p$ tensors in section 4. In section 5, we design a quantum tensor approximation algorithm based on classical truncated t-svd and then provide an application on context-aware multidimensional recommendation systems. In section 6, we conclude the paper.
2. Preliminaries

In section 2.1, we first introduce the concept of tensor, and notation used throughout the paper. In section 2.2, we review the concept of t-product and the t-svd algorithm proposed by Kilmer et al [1] in 2011. Then in section 2.3, we briefly retrospect the quantum singular value estimation algorithm (QSVE) [31] proposed by Kerenidis and Prakash.

2.1. Tensor background and notation

A tensor \( \mathcal{A} \in \mathbb{C}^{N_1 \times N_2 \times \cdots \times N_p} \) is a multidimensional array of data, where \( p \) is the order and \( (N_1, \ldots, N_p) \) is the dimension. The order of a tensor is the number of modes. For instance, \( \mathcal{A} \in \mathbb{C}^{N_1 \times N_2 \times N_3} \) is a third-order tensor of complex values with dimension \( N_i \) for mode \( i \), \( i = 1, 2, 3 \), respectively. In this sense, a matrix \( A \) can be considered as a second-order tensor, and a vector \( x \) is a tensor of order 1. For a third-order tensor, we use terms frontal slice \( \mathcal{A}(::, i) \), horizontal slice \( \mathcal{A}(i, ::) \) and lateral slice \( \mathcal{A}(::, :) \) (see figure 1). By fixing all indices but the last one, the result is a tube of size \( 1 \times 1 \times N_3 \), which is actually a vector. For example, \( \mathcal{A}(i, j, :) \) is the \( (i, j) \)-th tube of \( \mathcal{A} \).

Notation. In this paper, script letters are used to denote higher-order tensors \( (\mathcal{A}, \mathcal{B}, \cdots) \). Capital nonscript letters are used to represent matrices \( (A, B, \cdots) \), and vectors are written as boldface lower case letters \( (u, v, \cdots) \). DFT\( (u) \) refers to performing the discrete Fourier transform (DFT) on \( u \), which is computed by the fast Fourier transform represented in Matlab notation \( \text{fft}(u) \). The tensor after DFT along the third mode of \( \mathcal{A} \) is denoted by \( \hat{\mathcal{A}} \), i.e. \( \hat{\mathcal{A}} = \text{fft}(\mathcal{A}, [1], 3) \). Hence we have \( \mathcal{A} = \text{ifft}(\hat{\mathcal{A}}, [1], 3) \), which is the inverse of the above operation. We use \( A^{(i)} \) to denote the \( i \)-th frontal slice \( \mathcal{A}(::, i) \) for short, hence the \( m \)-th frontal slice of \( \hat{\mathcal{A}} \) is \( \hat{A}^{(m)} \). There are three types of product we would like to clarify here: \( u \odot v \) refers to the circular convolution between vectors \( u \) and \( v \), \( \odot \) is the element-wise product and \( \odot A \odot B \) represents the t-product between tensors \( A \) and \( B \).

2.2. The t-svd algorithm and t-product

In this subsection, we present the classical t-svd as well as its pseudocode, algorithm 1. For readability of the main text, all mathematical details are put in appendix A. Simply speaking, the t-svd of a tensor can be interpreted as the usual matrix svd in the Fourier domain, as can be seen in algorithm 1.

Theorem 1 tensor singular value decomposition (t-svd) [1]. For \( \mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3} \) its t-svd is given by

\[
\mathcal{A} = U^s S^s V^T,
\]

where \( U^s \in \mathbb{R}^{N_1 \times N_2 \times N_3} \) and \( V^s \in \mathbb{R}^{N_1 \times N_2 \times N_3} \) are orthogonal tensors, and every frontal slice of \( S^s \in \mathbb{R}^{N_1 \times N_2 \times N_3} \) is a diagonal matrix (see figure 2).

Algorithm 1. t-svd for third-order tensors [1]

Input: \( \mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3} \)
Output: \( U^s \in \mathbb{R}^{N_1 \times N_2 \times N_3}, S^s \in \mathbb{R}^{N_1 \times N_2 \times N_3}, V^s \in \mathbb{R}^{N_1 \times N_2 \times N_3} \)

\[
\hat{\mathcal{A}} = \text{fft}(\mathcal{A}, [1], 3);
\]
(Continued.)

for i = 1, …, N do
  \{U, S, V\} = svd(\(\hat{A}(\cdot; i, \cdot)\));
  \(\hat{U}(\cdot; i, \cdot) = U; \hat{S}(\cdot; i, \cdot) = S; \hat{V}(\cdot; i, \cdot) = V;\)
end for
\(U = \text{ifft}(\hat{U}(\cdot, \cdot, 3); \hat{S} = \text{ifft}(\hat{S}(\cdot, \cdot, 3); \hat{V} = \text{ifft}(\hat{\hat{V}(\cdot, \cdot, 3));\)

For an order-\(p\) tensor \(A \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_p}\), the frontal slices of \(A\) are referenced using linear indexing by reshaping the tensor into an \(N_1 \times N_2 \times N_3 \times \cdots \times N_p\) third-order tensor, then the \(i\)-th frontal slice is \(A(\cdot; i, \cdot)\) Using this representation, one version of MATLAB pseudocode of the t-svd algorithm for order-\(p\) tensors is provided below.

Algorithm 2. t-svd for order-\(p\) tensors [26]

\begin{itemize}
  \item \text{Input:} \(A \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_p}\), \(1 = N_1N_2 \cdots N_p\)
  \item for \(i = 1, \cdots, p\) do
    \item \(A = \text{fft}(A(\cdot, 
##, \cdot)\);
  \item end for
  \item for \(i = 1, \cdots, p\) do
    \item \(\{U, S, V\} = \text{svd}(\hat{A}(\cdot; i, \cdot));
    \item \(\hat{U}(\cdot; i, \cdot) = U; \hat{S}(\cdot; i, \cdot) = S; \hat{V}(\cdot; i, \cdot) = V;
    \item end for
    \item for \(i = 1, \cdots, p\) do
      \item \(\hat{U} = \text{ifft}(\hat{U}(\cdot, \cdot, 3); \hat{S} = \text{ifft}(\hat{S}(\cdot, \cdot, 3); \hat{V} = \text{ifft}(\hat{\hat{V}(\cdot, \cdot, 3));\)
  \end{itemize}

In the remainder of this subsection, we summarize some properties of t-svd, which will be used in the quantum tensor approximation algorithm to be developed in section 5.

In the t-svd literature, the diagonal elements of the tensor \(S\) are called the singular values of \(A\). Moreover, the \(l_2\) norms of the nonzero tubes \(S(i, i, :)\) are in descending order, i.e.

\[
\|S(0, 0, :)\|_2 \geq \|S(1, 1, :)\|_2 \geq \cdots \geq \|S(\min(N_1, N_2) - 1, \min(N_1, N_2) - 1, :\|_2.
\]

However, it should be noticed that the diagonal elements of \(S\) may be unordered and even negative due to the inverse DFT. Thus, the truncated t-svd method for data approximation or tensor completion is designed by truncating the diagonal elements of \(\hat{S}\) instead of \(S\), as the diagonal elements of the former are non-negative and ordered in descending order; see lemma 1.

Lemma 1 [1, 29]. Suppose the t-svd of the tensor \(A \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_p}\) is \(A = U^* S^\dagger V^\dagger\). Then we have

\[
A = \sum_{i=0}^{\min(N_1N_2) - 1} \hat{U}(\cdot, i, :)^* \hat{S}(i, i, :) \hat{V}(\cdot, i, :)^\dagger,
\]

where the matrices \(\hat{U}(\cdot, i, :)\) and \(\hat{V}(\cdot, i, :)\) and the vector \(\hat{S}(i, i, :)\) are regarded as third-order tensors. For \(1 \leq k \leq \min(N_1, N_2)\), define \(A_k = \sum_{i=0}^{k-1} \hat{U}(\cdot, i, :)^* \hat{S}(i, i, :) \hat{V}(\cdot, i, :)^\dagger\). Then

\[
A_k = \arg \min_{\hat{A} \in \mathcal{M}_k} \|A - \hat{A}\|_F,
\]

where \(\mathcal{M}_k = \{x \in \mathbb{R}^{N_1 \times k \times N_2}; y \in \mathbb{R}^{k \times N_2 \times N_2}\}\). Therefore, \(\|A - A_k\|_F\) is the theoretical minimal error, given by \(\|A - A_k\|_F = \sqrt{\sum_{i=k}^{\min(N_1N_2) - 1} \|S(i, i, :)\|_2^2}\).}

2.3. Quantum singular value estimation

In [31], Kerenidis and Prakash propose a quantum singular value estimation (QSVE) algorithm. They assume that the input data is stored in a classical binary tree data structure, as stated in the following lemma, such that the QSVE algorithm with access to this data structure can efficiently create superpositions of rows of the subsample matrix.
Lemma 2 [31], theorem 5.1. Consider a matrix $A \in \mathbb{R}^{N \times N}$ with $\tau$ nonzero entries. Let $A_i$ be its $i$-th row, and $s_k = \frac{1}{\|A_i\|} \|A_i\|$, $\|A_k\|$, $\|A_{k-1}\|$ $\|A_{k-1}\|^T$. There exists a data structure storing the matrix $A$ in $O(\tau \log^2(N_s N_2))$ space such that a quantum algorithm having access to this data structure can perform the mapping $U_2: |i\rangle |0\rangle \rightarrow |i\rangle |A_i\rangle$, for $i = 0, \cdots, N_1 - 1$ and $U_2^*: |j\rangle |0\rangle \rightarrow |s_j\rangle |j\rangle$, for $j = 0, \cdots, N_2 - 1$ in time $\text{polylog}(N_2)$.

The following lemma summarizes the main idea of the QSVE algorithm and its detailed description can be found in [31].

Lemma 3 [31], theorem 5.2. Let $A \in \mathbb{R}^{N \times N}$ and $x \in \mathbb{R}^{N}$ be stored in the data structure as mentioned in lemma 2. Let the singular value decomposition of $A$ be $A = \sum_{l=0}^{r-1} \sigma_l |u_l\rangle \langle v_l|$, where $r = \min(N_1, N_2)$. The input state $|x\rangle$ can be represented in the eigenstates of $A$, i.e. $|x\rangle = \sum_{l=0}^{N-1} \beta_l |\eta_l\rangle$. Let $\epsilon > 0$ be the precision parameter. Then there is a quantum algorithm, denoted as $U_{\text{QSVE}}$, that runs in time $O(\text{polylog}(N_2)/\epsilon)$ and achieves

\[ U_{\text{QSVE}}(|x\rangle |0\rangle) = \sum_{l=0}^{N-1} \beta_l |\eta_l\rangle \]

with probability at least $1 - 1/\text{poly}(N_2)$, where $\sigma_l$ is the estimated value of $\sigma_l$ satisfying $|\sigma_l - \sigma_l| \leq \epsilon \|A\|_F$ for all $l$.

Remark 1. With regard to the matrix $A$ stated in lemma 3, we can also choose the input state as $|A\rangle = \frac{1}{\|A\|} \sum_{l=0}^{r-1} \sigma_l |u_l\rangle |v_l\rangle$, corresponding to the vectorized form of the normalized matrix $\frac{A}{\|A\|}$ represented in the svd form. This representation of the input state is adopted in section 3. Note that we are able to express the state $|A\rangle$ in the above form even if the singular pairs of $A$ are not known. According to lemma 3, we can obtain $\sigma_l$, an estimate of $\sigma_l$, stored in the third register superposed with the singular vectors $\{u_l\}$, $\{v_l\}$ after performing $U_{\text{QSVE}}$, i.e. the output state is $\frac{1}{\|A\|} \sum_{l=0}^{r-1} \beta_l |u_l\rangle |v_l\rangle |\eta_l\rangle$, where $|\sigma_l - \sigma_l| \leq \epsilon \|A\|_F$ for all $l = 0, \cdots, r - 1$.

3. Quantum t-svd for third-order tensors

3.1. The algorithm

In this section, we first present our quantum t-svd algorithm, algorithm 3, for third-order tensors $A \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, then explain it in detail.

Assumption 1. Every frontal slice of $A$ is stored in a tree structure introduced in lemma 2.

Assumption 2. We can prepare the state

\[ |\mathcal{A}\rangle = \frac{1}{\|\mathcal{A}\|_F} \sum_{i=0}^{N_3-1} \sum_{j=0}^{N_2-1} \sum_{k=0}^{N_1-1} \mathcal{A}(i, j, k) |i\rangle |j\rangle |k\rangle \]

efficiently. Without loss of generality, we assume that $\|\mathcal{A}\|_F = 1$.

Algorithm 3. Quantum t-svd for third-order tensors

Input: tensor $A = (a_{ijk}) \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ prepared in a quantum state $|\mathcal{A}\rangle$, precision $\epsilon_{\text{SVE}}^m$, $m = 0, \cdots, N_1 - 1$, $r = \min(N_1, N_2)$.

Output: state $|\phi\rangle$.

1. Perform the QFT on the third register of the quantum state $|\mathcal{A}\rangle$ in (1) to get

\[ |\tilde{\mathcal{A}}\rangle = \frac{1}{\|\mathcal{A}\|} \sum_{i=0}^{N_3-1} \sum_{j=0}^{N_2-1} \sum_{k=0}^{N_1-1} \mathcal{A}(i, j, k) |i\rangle |j\rangle |k\rangle \]

2. Perform the controlled operation

\[ U \triangleq \sum_{i=0}^{N_3-1} L_{3i}^{(N)} \otimes |m\rangle \langle m| \]

on the state $|\tilde{\mathcal{A}}\rangle$ to obtain

\[ U|\tilde{\mathcal{A}}\rangle = \sum_{i=0}^{N_3-1} \sum_{j=0}^{N_2-1} \sum_{k=0}^{N_1-1} \mathcal{A}^{(m)}(i, j, k) \langle i| \langle j| \langle k| |m\rangle \langle m| \]

where $\mathcal{A}^{(m)}$ is the estimated value of $\mathcal{A}^{(m)}$, and the singular value decomposition of $\tilde{\mathcal{A}}^{(m)}$ is $\tilde{\mathcal{A}}^{(m)} = \sum_{i=0}^{N_3-1} \beta_i^{(m)} |i\rangle |\tilde{\eta}_i\rangle |\tilde{\eta}_i\rangle$. 

3. Perform the inverse QFT on the last register of (4) and output the state $|\phi\rangle$ expressed as

\[ \frac{1}{\|\tilde{\mathcal{A}}\|} \sum_{i=0}^{N_3-1} \sum_{j=0}^{N_2-1} \sum_{k=0}^{N_1-1} \beta_i^{(m)} e^{-i \frac{2\pi}{N_3} (m)} |i\rangle |\tilde{\eta}_i\rangle |\tilde{\eta}_i\rangle \]

where $\omega = e^{2\pi i/3}$. 

The quantum circuit of algorithm 3 is shown in figure 3, where the blocks $U_{\text{SVE}}$, $m = 0, \cdots, N_3 - 1$, are illustrated in figure 4.
Before illustrating the algorithm, we first interpret the final quantum state \(|\phi\rangle\). Similar to the quantum singular value decomposition for matrices \([33]\) that the output allows singular values and associated singular vectors to be revealed in a quantum form, the output state \(|\phi\rangle\) in our algorithm also finds the estimated values of \(\hat{l}_m^s\), which are stored in the third register, in superposition with corresponding singular vectors. Although the singular values of the tensor \(\mathcal{A}\) are defined as 
\[
\sum_{m=0}^{N_1-1} \omega^{-kn} \hat{l}_m^s
\]
according to algorithm 1 for the classical t-svd, the singular values of \(A_m^s\), \(\sigma_f^s\), have wider applications than the singular values of \(\mathcal{A}\), \(\sigma_f^s\). For example, some low-rank tensor completion problems are solved by minimizing the tensor nuclear norm of the tensor, which is defined as the sum of all the singular values of \(A_m^s\) \([3, 5]\). Moreover, the theoretical minimal error truncation is also based on the singular values of \(A_m^s\); see lemma 1. Therefore, in algorithm 3, we estimate the values of \(\hat{l}_m^s, m = 0, \cdots, N_1-1\), and store them in the third register of the final state \(|\phi\rangle\) for future use. Furthermore, in terms of the circulant matrix \(\text{circ}(\mathcal{A})\) defined in definition 1, 
\[
\sum_{m=0}^{N_1-1} \omega^{-kn} |\phi_f^s(m)\rangle\langle \phi_f^s(m)|
\]
is the right singular vector corresponding to its singular value \(\hat{l}_m^s\). Similarly, the corresponding left singular vector is 
\[
\sum_{m=0}^{N_1-1} \omega^{-kn} |\phi_f^s(m)\rangle\langle \phi_f^s(m)|
\]
Next, we give a detailed explanation on Step 2. We first rewrite the state in (2) for further use. For every fixed \(m\), the unnormalized state
\[
\frac{1}{\sqrt{N_3}} \sum_{k,l=0}^{N_3-1} \omega^{km} |a_{kl}| \langle f |\langle f |
\]
Figure 3. Circuit for algorithm 3. \(U_A\) is the unitary operator for preparing the state \(|\mathcal{A}\rangle\). The QFT is denoted by \(F\). \(N_i = 2^n, i = 1, 2, 3\). The blocks \(U_{SV^nE}^i\) are further illustrated in figure 4.

Figure 4. Circuit for \(U_{SV^nE}^m\), \(m = 0, \cdots, N_1-1\). The initial state of register c and d is \(A^m\). \(U_m^r\) refers to \(U_{SV^nE}^m\). \(U_f^m\) is a unitary operator implemented through oracle with a computable function \(f_m(x)\). The notation is further explained in the proof of theorem 2 in appendix B.
in (2) corresponds to the matrix

\[
\hat{A}^{(m)} = \frac{1}{\sqrt{N_3}} \sum_{i,j,k} \omega^{km} a_{ijk} |j\rangle \langle j|,
\]

namely, the \(m\)-th frontal slice of the tensor \(\hat{A}\). Normalizing the state in (6) produces a quantum state

\[
|\hat{A}^{(m)}\rangle = \frac{1}{\sqrt{N_3 \|A^{(m)}\|_F}} \sum_{i,j,k} \omega^{km} a_{ijk} |j\rangle |j\rangle.
\]

Therefore, the state \(|\hat{A}\rangle\) in (2) can be rewritten as

\[
|\hat{A}\rangle = \sum_{m=0}^{N_3-1} |\hat{A}^{(m)}\rangle |A^{(m)}\rangle_{cd} |m\rangle^c.
\]

In Step 2, we utilize a controlled operation \(U\) defined in (3) to estimate the singular values of \(\hat{A}^{(m)}\) parallley, \(m = 0, \cdots, N_3 - 1\). Due to the quantum parallelism, the operator \(U\) performed on the superposition state \(|\hat{A}\rangle\) is thus equivalent to \(U_{SVE}^{(m)}\) performed on each of the components \(|\hat{A}^{(m)}\rangle\) as a single input. That is,

\[
U |\hat{A}\rangle = \sum_{m=0}^{N_3-1} |\hat{A}^{(m)}\rangle \|U \langle A^{(m)}\rangle_{cd} |m\rangle^c.
\]

Next, we focus on the result of \(U_{SVE}^{(m)}|\hat{A}^{(m)}\rangle\) in (10). The state \(|\hat{A}^{(m)}\rangle\) can be rewritten in the form

\[
|\hat{A}^{(m)}\rangle = \sum_{l=0}^{r-1} |\hat{A}_l^{(m)}\rangle \|\hat{A}_l^{(m)}\rangle_{cd} |l\rangle^d,
\]

where \(|\hat{A}_l^{(m)}\rangle\) is the rescaled singular value of \(\hat{A}^{(m)}\). The following theorem describes \(U_{SVE}^{(m)}\), a modified quantum singular value estimation process on each matrix \(\hat{A}^{(m)}\) utilizing the corresponding input \(|\hat{A}^{(m)}\rangle\) represented in (11).

**Theorem 2.** Given every frontal slice of the original tensor \(A\) stored in the data structure (Lemma 2), there is a quantum algorithm, denoted as \(U_{SVE}^{(m)}\), that uses the input \(|\hat{A}^{(m)}\rangle\) in (11) and outputs the state

\[
U_{SVE}^{(m)}|\hat{A}^{(m)}\rangle = \frac{1}{\|\hat{A}^{(m)}\|_F} \sum_{l=0}^{r-1} \hat{A}_l^{(m)} |\hat{A}_l^{(m)}\rangle_{cd} |l\rangle^d,
\]

with probability at least \(1 - 1/\text{poly}(N_3)\), where \((\hat{\sigma}_l^{(m)}, \hat{u}_l^{(m)}, \hat{v}_l^{(m)})\) is the singular triplet of the matrix \(\hat{A}^{(m)}\) in (7), and \(\epsilon_{SVE}\) is the precision such that \(\|\hat{A}_l^{(m)} - \hat{\sigma}_l^{(m)} \hat{u}_l^{(m)} \hat{v}_l^{(m)}^\dagger\|_F \leq \epsilon_{SVE}\|\hat{A}^{(m)}\|_F\) for all \(l = 0, \cdots, r - 1\). For tensor \(A\) with same dimension \(N\) on every order, the running time to implement \(U_{SVE}^{(m)}\) is \(O(N \text{polylog} N / \epsilon_{SVE})\).

**Proof.** See appendix B.

Actually, the process \(U_{SVE}^{(m)}\) proposed in theorem 2 is quite different from the original QSVE technique introduced in lemma 3. In theorem 2, it is proved that we can estimate the singular values of \(\hat{A}^{(m)}\) on the condition that each frontal slice of the original tensor \(A^{(k)}\) is stored in the binary tree. The proof of theorem 2 presents a detailed illustration of the procedure of \(U_{SVE}^{(m)}\), and the circuit shown in figure 4 can help to understand it.

Thus after Step 2, the state in (10) becomes the state in (4) based on theorem 2.

Our quantum t-svd algorithm can be used as a subroutine of other algorithms, that is, it is suitable for some specific applications where the singular values of \(\hat{A}^{(m)}\) are used. For example, some third order tensor completion problems can be efficiently solved by extracting the singular values of \(\hat{A}^{(m)}\) and only keeping the greater ones. Moreover, some context-aware recommendation systems also utilize tensor factorizations, such as the truncated t-svd [3] and the truncated HOSVD [32]. See more details in section 5.

**Remark 2.** Note that the input of \(U_{SVE}^{(m)}\) is \(|\hat{A}^{(m)}\rangle\) instead of an arbitrary quantum state commonly used in some quantum svd algorithms [34]. This input fits our quantum t-svd algorithm better since it keeps the entire singular information \((\hat{\sigma}_l^{(m)}, \hat{u}_l^{(m)}, \hat{v}_l^{(m)})\) completely, thus our algorithm can output a quantum state whose representation is similar to the matrix svd. Another consideration is that we do not need the information unrelated to the tensor \(A\) (e.g. an arbitrary state) to be involved in our algorithm.
3.2. Complexity analysis
For simplicity, we consider a tensor $A \in \mathbb{R}^{N \times N \times N}$ with the same dimensions on each mode. In Steps 1 and 3 of algorithm 3, performing the QFT or the inverse QFT parallelly on the third register of the state $|A\rangle$ has the complexity of $\mathcal{O}(\log N^2)$, compared with the complexity $\mathcal{O}(N^3 \log N)$ of the DFT performed on $N^2$ tubes of the tensor $A$ in the classical t-svd algorithm. Moreover, in the classical t-svd, the complexity of performing the matrix svd (Step 2 of algorithm 1) for all frontal slices of $A$ is $\mathcal{O}(N^4)$. In contrast, in our quantum t-svd algorithm, this step is accelerated by theorem 2 (the modified QSVE) whose complexity is $\mathcal{O}(N\text{polylog}(N))$ on each frontal slice $A^{(m)}$. Since we perform this modified QSVE on each $A^{(m)}$ parallelly, $m = 0, \cdots, N - 1$, the running time of Step 2 is still $\mathcal{O}(N\text{polylog}(N))$. Therefore, the total computational complexity of algorithm 3 is $\mathcal{O}(N\text{polylog}(N))$.

4. Quantum t-svd for order-$p$ tensors
Following a similar procedure, we can extend the quantum t-svd for third-order tensors to order-$p$ tensors easily.

We assume that the quantum state $|A\rangle$ corresponding to the tensor $A \in \mathbb{R}^{N_1 \times \cdots \times N_p}$ can be prepared efficiently, where $N_i = 2^n$ with $n_i$ being the number of qubits on the corresponding mode and

$$|A\rangle = \sum_{j_1=0}^{N_1-1} \cdots \sum_{j_p=0}^{N_p-1} A(i_1, \cdots, i_p) |i_1, \cdots, i_p\rangle.$$  \hspace{1cm} (13)

Next, we perform the QFT on the third to the $p$-th mode of the state $|A\rangle$, and then use $|m\rangle$ to denote $|m_1\rangle \cdots |m_p\rangle$, i.e. $m = m_1 \Pi_{j=1}^p N_j + m_2 \Pi_{j=2}^p N_j + \cdots + m_p$. The value of $m$ ranges from 0 to $\ell - 1$, where $\ell = N_2N_4 \cdots N_p + N_3N_5 \cdots N_p + \cdots + N_p$, specifically, $\ell = (N^{p-1} - N)/(N - 1)$ when $N_3 = \cdots = N_p = N$. Then we obtain

$$|\hat{A}\rangle = \frac{1}{\sqrt{\ell}} \sum_{m=0}^{\ell-1} \sum_{i_1, \cdots, i_p} \omega^{i_1m_1} A(i_1, \cdots, i_p) |i_1, \cdots, i_p\rangle |m\rangle.$$ \hspace{1cm} (14)

Let the matrix

$$\hat{A}^{(m)} = \frac{1}{\sqrt{\ell}} \sum_{i_1, \cdots, i_p} \sum_{j_1, \cdots, j_p} \omega^{j_1m_1} A(i_1, \cdots, i_p) |i_1\rangle |j_1\rangle$$

and perform the modified QSVE on $\hat{A}^{(m)}$, $m = 0, \cdots, \ell - 1$, parallelly using the same strategy described in section 3.1, we can get the state

$$|\psi_p\rangle = \sum_{m=0}^{\ell-1} \left( \sum_{i=0}^{\ell-1} \hat{\xi}_i^{(m)} |\xi_i^{(m)}\rangle |\hat{\eta}_i^{(m)}\rangle \right) |m\rangle.$$ \hspace{1cm} (15)

after Step 2.

Finally, we recover the $|m_1\rangle \cdots |m_p\rangle$ expression and perform the inverse QFT on the $p$-th to the third register, obtaining the final state

$$|\phi_p\rangle = \frac{1}{\sqrt{\ell}} \sum_{m=0}^{\ell-1} \sum_{i_1, \cdots, i_p} \omega^{i_1m_1} A_1^{(m)} |i_1\rangle |i_2\rangle |i_3\rangle$$ \hspace{1cm} (16)

corresponding to the quantum t-svd of order-$p$ tensor $A$.

Algorithm 4. Quantum t-svd for order-$p$ tensors

**Input:** tensor $A \in \mathbb{R}^{N_1 \times \cdots \times N_p}$ prepared in a quantum state, precision $\epsilon_{\text{SVT}}^{(m)}$, $m = 0, \cdots, \ell - 1$.

**Output:** state $|\phi_p\rangle$.

1. Perform the QFT parallelly from the third to the $p$-th register of quantum state $|A\rangle$, obtain the state $|\hat{A}\rangle$.
2. Perform the modified QSVE for each matrix $\hat{A}^{(m)}$ with precision $\epsilon_{\text{SVT}}^{(m)}$ parallelly, $m = 0, \cdots, \ell - 1$, by using the controlled-U$\text{SVT}$ acting on the state $|\hat{A}\rangle$, to obtain the state $|\psi_p\rangle$.
3. Perform the inverse QFT parallelly from the third to the $p$-th register of the above state and output the state $|\phi_p\rangle$. 

8
For order-\(p\) tensor \(\mathcal{A} \in \mathbb{R}^{N_1 \times \cdots \times N_p}\), compared with the classical t-svd algorithm [26] whose time complexity is \(O(N^{p+1})\), our algorithm output a quantum state with the classical t-svd information encoded, and the time complexity of our quantum algorithm is \(O(N \text{ polylog}(N))\) followed by a similar analysis.

5. Application to context-aware POI recommendation systems

In this section, we apply algorithm 3 to another quantum algorithm, algorithm 5, which implements a classical machine learning task: context-aware POI recommendation systems.

In point-of-interest (POI) recommendations, a user’s preference for a POI, such as restaurants and sightseeing sites, is seriously influenced by context, such as the time slot in a day, his/her current location, etc. Therefore, many works focus more on integrating multiple contexts in a more efficient manner to improve user experience. Tensors are a natural choice to model high-order contextual information, e.g., a user’s rating for different POIs can be encoded in the preference tensor \(T\) in which the three modes represent locations, POIs, and time frames respectively. The classical (non-quantum) context-aware POI recommendation systems utilizing various tensor decomposition methods have been tested by experiment to have better result (both accuracy and execution time) than non-contextual modeling [35, 36]. However, these methods are computational expensive due to the high cost of tensor decomposition. In fact, the classical 3D recommendation systems algorithm based on tensor decomposition, such as the truncated t-svd with complexity \(O(kn^3 + N^3 \log N)\) [3], or the truncated HOSVD (T-HOSVD) with complexity \(O(3kn^3)\) [32], where \(k\) is the truncation rank and \(N\) is the dimension of the preference tensor. Considering the effectiveness and high cost of context-aware POI recommendation systems, we propose a quantum context-aware POI recommendation systems algorithm (QC-POI) with lower complexity \(O(N \text{ polylog}N)\) for some suitable parameters, and the result of this algorithm is the classical information of POI recommendation indices.

The problem of the context-aware POI recommendations can be stated as follows. Suppose there is a hidden preference tensor \(T\) that encodes the preference information of a given user under two contexts, e.g., locations and time. In practical applications, only a part of the entries of \(T\) can be observed. Actually, users typically engage with only a small subset of POIs and a considerable amount of possible interactions stays unobserved. For example, a person may be simply unaware of existing alternatives for the POIs of his/her choice. Finding out those entries helps to make better predictions. We denote the tensor whose entries are observed ratings as \(\mathcal{A}\), which is sparse in general. Our goal is to predict these unobserved triples (location, POI, time) and recommend some of his/her favorite POIs among all contexts based on those comparatively high predicted values. For example, Alice has visited several cities (locations) in America, and her ratings of different POIs when she was in these cities are encoded in a tensor \(T\) whose three modes are cities, POIs, time respectively. For some reasons, she only scores a part of POIs of these cities, denoted as \(\mathcal{A}\), and our task is to provide her favorite POIs among all cities and time slots based on the truncated t-svd of the tensor \(\mathcal{A}\).

In this application, we make two assumptions. Firstly, we assume that tensor the \(T\) is of low tubal-rank. Secondly, the original tensor \(T\) has several dominating entries. In fact, this low tubal-rank assumption is also adopted in the classical truncated t-svd data completion problem [2, 3, 37]. For the second assumption, in real situations, it is very likely that a user has some favorite POIs among all contexts (dominating entries).

Our QC-POI algorithm, algorithm 5, includes three processes: quantum t-svd, quantum state projection and quantum measurement. After algorithm 3 and the quantum state projection process, we can obtain the state \(|\mathcal{A}_{\mathcal{A}_{\mathcal{A}_s}}\rangle\) corresponding to an approximation of the hidden preference tensor \(T\) under certain conditions; similar conclusion could be seen in [19], theorem 1. Thus, previously unobserved values of triples corresponding to this user’s favourite POIs in the hidden preference tensor \(T\) might be boosted after projected t-svd on observed tensor \(\mathcal{A}\). As \(\mathcal{A}_{\mathcal{A}_s}\) is non-sparse in general, we can predict the missing entries based on the relatively high predicted values in \(\mathcal{A}_{\mathcal{A}_s}\), and provide POI recommendations by measuring the state \(|\mathcal{A}_{\mathcal{A}_s}\rangle\) in the computational basis. The complexity of our QC-POI of getting a good recommendation is \(O(N \text{ polylog}N)\) for suitable chosen parameters, see the analysis in the paragraph below algorithm 5. The output is a POI recommendation index for this user under all locations and time slots.

In fact, our quantum 3D recommendation systems algorithm borrows the idea of quantum recommendation systems for matrices proposed by Kerenidis and Prakash [31]. For recommendation systems modeled by an \(m \times n\) preference matrix, Kerenidis and Prakash designed a quantum algorithm that offers recommendations by just measuring the quantum state representing an approximation of the hidden preference matrix obtained by truncated matrix SVD.

The following is a summary of the steps of algorithm 5. We first follow the Steps 1 and 2 of algorithm 3, then perform the quantum projection with a pre-specified threshold \(\sigma\) on state (4). Specifically, for state (4), we apply the unitary operator \(V\) on the register \(a\) and an ancillary register \(|0\rangle\) that maps \(|t\rangle^n |0\rangle \rightarrow |t\rangle^n |1\rangle\) if \(t < \sigma\) and \(|t\rangle^n |0\rangle \rightarrow |t\rangle^n |0\rangle\) otherwise. Therefore, after Step 2 of algorithm 3, we get
\[
\sum_{m=0}^{N_0-1} \left( \sum_{l:l_0 \leq l < m} \delta_l^{(m)} |\tilde{a}_l^{(m)}\rangle \langle |\tilde{b}_l^{(m)}\rangle \langle d |\tilde{c}_l^{(m)}\rangle |0\rangle + \sum_{l:l_0 > m} \delta_l^{(m)} |\tilde{a}_l^{(m)}\rangle \langle |\tilde{b}_l^{(m)}\rangle \langle d |\tilde{c}_l^{(m)}\rangle |1\rangle \right) |m\rangle^r.
\] (17)

Next, we apply the inverse modified QSVE on state (17) and discard the register \(a\). Then we measure the third register of (17), and postselect the outcome \(|0\rangle\), obtaining
\[
\frac{1}{\alpha} \sum_{m,l:h_0 \leq |m\rangle \leq \sigma} \delta_l^{(m)} |\tilde{a}_l^{(m)}\rangle \langle |\tilde{b}_l^{(m)}\rangle \langle d |\tilde{c}_l^{(m)}\rangle |m\rangle^r,
\] (18)
where
\[
\alpha = \sqrt{\sum_{m,l:h_0 \leq |m\rangle \leq \sigma} (\delta_l^{(m)})^2}.
\] (19)

The probability that we obtain the outcome \(|0\rangle\) is
\[
\frac{\sum_{m,l:h_0 \leq |m\rangle \leq \sigma} (\delta_l^{(m)})^2}{\sum_{m=0}^{N_0-1} \sum_{s=0}^{\sigma} \sum_{l=0}^{l_0} (\delta_l^{(m)})^2} = \frac{\|\tilde{A}_{\sigma}\|_F^2}{\|\tilde{A}\|_F^2} = \frac{\alpha^2}{\|\tilde{A}\|_F^2},
\] (20)
since the Frobenius norm is unchanged under the Fourier transform. Tensor \(\tilde{A}_{\sigma}\) denotes the tensor whose \(m\)-th frontal slice is \(\tilde{A}^{(m)}_{\sigma}\), obtained by truncating \(A^{(m)}\) with threshold \(\sigma\), and \(\tilde{A}_{\sigma}\) is the inverse QFT of \(\tilde{A}_{\sigma}\). Based on amplitude amplification, we have to repeat the measurement \(O(1/\alpha)\) times in order to ensure the success probability of getting the outcome \(|0\rangle\) is close to 1. Thus, the complexity of getting the state 18 is \(O(N \text{ polylog} N/\alpha)\). Combining (9) with (11), we find that the state (18) can be seen as an approximation of \(|\tilde{A}\rangle\).

In Step 4, we perform the inverse QFT on state (18), obtaining the final state denoted as
\[
|A_{\sigma}\rangle = \frac{1}{\sqrt{N_0}} \sum_{m=0}^{N_0-1} \sum_{l: \delta_l^{(m)} \geq \sigma} \delta_l^{(m)} \omega^{-m} |\tilde{a}_l^{(m)}\rangle \langle |\tilde{b}_l^{(m)}\rangle \langle d |\tilde{c}_l^{(m)}\rangle |r\rangle^r,
\] (21)
which is an approximation of the state \(|A\rangle\) in the classical counterpart algorithm [3]. The effectiveness of the classical counterpart of algorithm 3 has been tested by numerical experiments; see more detail in remark 3.

**Remark 3.** The classical counterpart of algorithm 5 corresponds to a tensor completion method proposed in [3] and its effectiveness has been verified by numerical experiments. In what follows we briefly summarize this algorithm. For an \(N_1 \times N_2 \times N_3\) tensor \(A\), we use algorithm 1 to get \(\tilde{A}, \tilde{F}, \tilde{S}, \tilde{V}\). The total number of diagonal entries of \(\tilde{S}\) is \(rN_0\), where \(r = \min\{N_1, N_2, N_3\}\). After sorting them from the largest to the smallest, suppose these entries decay rapidly or small values are the majority, there exists a number \(k\) such that the top \(k\) diagonal entries of \(\tilde{S}\), denoted as \(\tilde{S}_k\), and setting other entries to be 0 result in a good approximation. Thus, the approximate tensor is \(\tilde{A}_k = \tilde{U}_k^{\dagger} \tilde{S}_k^{\dagger} \tilde{V}_k^{\dagger}\), where \(\tilde{U}_k\), \(\tilde{S}_k\), and \(\tilde{V}_k\) are the inverse DFT of \(\tilde{U}_k\), \(\tilde{S}_k\), and \(\tilde{V}_k\), respectively. According to the simulation results in [3], this approximation method has the lowest relative square error (RSE) compared with the other two compression methods, and it has good performance for video datasets from both stationary and non-stationary cameras.

**Algorithm 5.** Quantum context-aware POI recommendation systems

**Input:** The observed tensor \(A = (a_{ijk}) \in \mathbb{R}^{N_0 \times N_1 \times N_2}\) satisfying Assumption 1, threshold \(\sigma\), \(r = \min\{N_0, N_1, N_2\}\).

**Output:** a recommendation index.

1: Follow the Steps 1 and 2 of algorithm 3.
2: Apply the unitary operator \(V\) on the register \(a\) and an ancillary register \(|0\rangle\) that maps \(|r\rangle\langle 0| \rightarrow |r\rangle\langle 1|\) if \(t < \sigma\) and \(|r\rangle\langle 0| \rightarrow |r\rangle\langle 0|\) otherwise, obtaining the state in (17).
3: Perform the inverse modified QSVE on (17), discard the register \(a\) and postselect the outcome \(|0\rangle\) to obtain state in (18).
4: Perform the inverse QFT on state (18) to obtain \(|A_{\sigma}\rangle\) in (21).
5: Measure the output state in (21) in the computational basis to get a recommendation index.

In the final step, we measure the output quantum state \(|A_{\sigma}\rangle\) in the computational basis in order to extract some classical information (recommendation indices) from the approximate quantum state (21). A good recommendation means that the amplitude of the measurement outcome corresponds to that dominating entries in the hidden preference tensor. Next we give a rough estimation on the complexity of obtaining a good recommendation by algorithm 5. Denote the sum of the squares of the dominating entries of tensor \(A_{\sigma}\) as \(\nu\) (corresponding to the large amplitudes of the state \(|A_{\sigma}\rangle\)). According to above analysis, the total cost of first four steps of algorithm 5 is \(O(N \text{ polylog} N \|A\|_F^2/\alpha)\), where \(\alpha\) is the Frobenius norm of tensor \(A_{\sigma}\) and it is defined...
in (19). Then after Step 5, we need to repeat the measurement \( O(\alpha / \sqrt{v}) \) times in order to ensure the probability of getting a good POI recommendation index is close to 1. Therefore, the complexity of our QC-POI algorithm is \( O(N \text{ polylog} N \| A \| F / \sqrt{v}) \) when \( N_1 = N_2 = N_3 = N \). According to that dominating entries assumption, the complexity of algorithm 5 could be \( O(N \text{ polylog} N) \) for large \( N \). The measurement outcome corresponds to a triplet (location, POI, time slot) whose second entry is the recommended POI index, and this algorithm achieves a polynomial speedup compared with the classical 3D recommendation systems algorithm.

We summarize the main steps of this algorithm in algorithm 5.

6. Conclusion

In this paper, we propose a quantum t-svd algorithm for third-order tensors with complexity \( O(N \text{ polylog}(N)) \). The key tools that accelerate this process are quantum Fourier transform and quantum singular value estimation, then we extend this third-order tensor algorithm to order- \( p \) tensors. Moreover, based on the quantum t-svd algorithm, a quantum tensor approximation algorithm is proposed, which is applied to context-aware 3D recommendation systems.

Appendix A. Relevant definitions and results of the classical t-svd algorithm

In this section, we provide the relevant definitions and results of the classical t-svd algorithm, algorithm 1 in the main text.

Definition 1 [1] circulant matrix. Given a vector \( u \in \mathbb{R}^N \) and a tensor \( B \in \mathbb{R}^{N_1 \times N_2 \times N_3} \) with frontal slices \( B^{(l)} \), \( l = 0, \ldots, N_3 - 1 \), the matrices \( \text{circ}(u) \) and \( \text{circ}(B) \) are defined as

\[
\text{circ}(u) = \begin{bmatrix}
    u_0 & u_{N-1} & \cdots & u_1 \\
    u_1 & u_0 & \cdots & u_2 \\
    \vdots & \vdots & \ddots & \vdots \\
    u_{N-1} & u_{N-2} & \cdots & u_0
\end{bmatrix}
\]

\[
\text{circ}(B) = \begin{bmatrix}
    B^{(0)} & B^{(N_3-1)} & \cdots & B^{(1)} \\
    B^{(1)} & B^{(0)} & \cdots & B^{(2)} \\
    \vdots & \vdots & \ddots & \vdots \\
    B^{(N_3-1)} & B^{(N_3-2)} & \cdots & B^{(0)}
\end{bmatrix}
\]

respectively.

Definition 2 circular convolution. Let \( u, v \in \mathbb{R}^N \). The circular convolution between \( u \) and \( v \) produces a vector \( x \) of the same size, defined as

\[
x \equiv u \otimes v \triangleq \text{circ}(u)v.
\]

As a circulant matrix can be diagonalized by means of the discrete Fourier transform (DFT), from definition 2, we have \( \text{DFT}(x) = \text{diag}(\text{DFT}(u))\text{DFT}(v) \), where diag \( (u) \) returns a square diagonal matrix with elements of the vector \( u \) on the main diagonal. As a result, the circular convolution between two vectors in definition 2 is better understood in the Fourier domain, which is given by the following result.

Theorem 3 Cyclic Convolution theorem [38]. Given \( u, v \in \mathbb{R}^N \), let \( x = u \otimes v \) as defined in definition 2. We have

\[
\text{DFT}(x) = \text{DFT}(u) \odot \text{DFT}(v),
\]

where \( \odot \) denotes the element-wise product.

If a tensor \( A \in \mathbb{R}^{N_1 \times N_2 \times N_3} \) is considered as an \( N_1 \times N_2 \) matrix whose \((i, j)\)-th entry is a tube of dimension \( N_3 \), i.e. \( A(i, j, :) \), then based on definition 2, the t-product between tensors is defined as follows.

Definition 3 t-product [1]. Let \( M \in \mathbb{R}^{N_1 \times N_2 \times N_3} \) and \( N \in \mathbb{R}^{N_1 \times N_2 \times N_3} \). The t-product of \( M \) and \( N \), i.e. \( A \triangleq M \ast N \), is an \( N_1 \times N_2 \times N_3 \) tensor \((i, j, :)\)-th tube is

\[
A(i, j, :) = \sum_{k=0}^{N_3-1} M(i, k, :) \otimes N(k, j, :)
\]

for all \( i = 1, \ldots, N_1 \) and \( j = 1, \ldots, N_2 \).
Similar as the circular convolution in definition 2, the t-product in definition 3 can be better interpreted in the Fourier domain. Specifically, let \( \hat{A} \) be the tensor whose \((i, j)\)-th tube is \( \text{DFT}(A(i, j, :)) \). Then by theorem 3 and definition 3, we have

\[
\hat{A}(i, j, :) = \sum_{k=0}^{N_0-1} \hat{\mathcal{M}}(i, k, :) \odot \hat{\mathcal{N}}(k, j, :),
\]

which is the Fourier counterpart of equation (A.2). Interestingly, for a fixed index \( l \) in the third mode, equation (A.3) is equivalent to \( \hat{A}^{(l)} = \hat{\mathcal{M}}^{(l)} \hat{\mathcal{N}}^{(l)} \), which is the conventional matrix product. This nice equivalence relations between t-product and matrix multiplication (in the Fourier domain) are summarized in the following theorem.

**Theorem 4** [1] For tensors \( \mathcal{M} \in \mathbb{R}^{N_1 \times N_2 \times N_3} \) and \( \mathcal{N} \in \mathbb{R}^{N_2 \times N_3 \times N_4} \), the equivalence relation

\[
\mathcal{A} = \mathcal{M}^* \mathcal{N} \iff \hat{A}^{(l)} = \hat{\mathcal{M}}^{(l)} \hat{\mathcal{N}}^{(l)}
\]

holds for \( l = 0, \ldots, N_3 - 1 \), where \( \hat{A}^{(l)} \) is the \( l \)-th frontal slice of \( \hat{A} \).

By the equivalence relation given in theorem 4 above, the t-svd defined in theorem 1 in the main text can be interpreted in as the matrix SVD in the Fourier domain, as reflected in algorithm 1. In what follows, we list some definitions used in theorem 1 and algorithm 1.

Firstly, tensor transpose operation is used in theorem 1, which is defined as follows.

**Definition 4 tensor transpose**[1]. The transpose of a tensor \( \mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3} \), denoted by \( \mathcal{A}^T \), is obtained by transposing all the frontal slices and then reversing the order of the transposed frontal slices 1 through \( N_3 - 1 \).

The tensor transpose defined in definition 4 has the same property as the matrix transpose, i.e. \((A^* B)^T = B^* A^T \).

Secondly, orthogonal tensors are used in theorem 1, whose definition is given below.

**Definition 5 orthogonal tensor**[1]. A tensor \( \mathcal{U} \in \mathbb{R}^{N_1 \times N_2 \times N_3} \) is an orthogonal tensor if it satisfies

\[
\mathcal{U}^T \mathcal{U} = \mathcal{I} \quad \text{and} \quad \mathcal{U} \mathcal{U}^T = \mathcal{I},
\]

where \( \mathcal{I} \in \mathbb{R}^{N_1 \times N_1} \) is an identity tensor, in other words, its first frontal slice \( I(0) \) is an \( N_1 \times N_1 \) identity matrix and all the other frontal slices are zero matrices.

Finally, we give the definition of tensor Frobenius norm as it is quite useful for tensor approximation.

**Definition 6 tensor Frobenius norm**[1]. The Frobenius norm of a third-order tensor \( \mathcal{A} = (a_{ijk}) \) is defined as

\[
\| \mathcal{A} \|_F = \sqrt{\sum_{i,j,k} |a_{ijk}|^2}.
\]

Similar to orthogonal matrices, the orthogonality defined in definition 5 preserves the Frobenius norm of a tensor, i.e. given an orthogonal tensor \( \mathcal{Q} \), we have \( \| \mathcal{Q}^* \mathcal{A} \|_F = \| \mathcal{A} \|_F \). Moreover, when the tensor is second-order, definition 5 coincides with the definition of orthogonal matrices.

**Appendix B. The proof of theorem 2**

Before proving theorem 2, we would like to sketch the proof first. According to Assumption 1, each frontal slice \( A^{(k)} \) is stored in the binary tree structure, hence based on the proof of lemma 5 in [31], the states \(| \mathcal{A}(i, :, k) \rangle \), corresponding to the \( i \)-th row of \( A^{(k)} \), can be prepared efficiently by operators \( P^{(k)} \). Based on these operators, two new isometries \( P_\mathcal{A} \) and \( Q_\mathcal{A} \), are constructed in order to perform QSVE on \( \hat{\mathcal{A}}^{(0)} \). Moreover, the input of our modified QSVE is also different from that in [31]. The detail of the proof is given below.

**Proof**. Since every \( A^{(k)} \), \( k = 0, \ldots, N_3 - 1 \), is stored in the binary tree structure, the quantum computer can perform the following mapping in \( \mathcal{O}(\text{polylog}(N_3)) \) time, as shown in theorem 5.1 in [31]:

\[
U_{\mathcal{P},\mathcal{A}}: |i\rangle |0\rangle \rightarrow |i\rangle |\mathcal{A}(i, :, k)\rangle = \frac{1}{\| \mathcal{A}(i, :, k) \|_2} \sum_{j=0}^{N_0-1} a_{ijk} |j\rangle |j\rangle,
\]

where \( \mathcal{A}(i, :, k) \) is the \( i \)-th row of \( A^{(k)} \).
Define the degenerate operator $P_k \in \mathbb{R}^{N \times N}$ related to $U_{P_k}$ as
$$P_k: |i\rangle \rightarrow |i\rangle \Lambda(i, :; k).$$
That is,
$$P_k = \sum_{i=0}^{N-1} |i\rangle \Lambda(i, :; k) \langle i|.$$  \hfill (B.2)

Based on the efficiently implemented operators $P_k$ and $U_{P_k}$, we define another operator
$$U_{P_k} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{km} U_{P_k}.$$  \hfill (B.3)

It can be easily followed that the operator $U_{P_k}$ achieves the state preparation of the rows of the matrix $\Lambda^{(m)}$, i.e.,
$$U_{P_k} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{km} |\Lambda(i, :; k)\rangle \langle i| \rangle \langle i| = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{km} |\Lambda(i, :; k)\rangle \langle i|.$$  \hfill (B.4)

It is also true for another operator $U_{P_k}^2 = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{km} |\Lambda(i, :; k)\rangle \langle i|\Lambda(i, :; k)\rangle \langle i|.$  \hfill (B.5)

Next, we define the mapping
$$U_{Q_m}^* |0\rangle \langle j| \rightarrow |s_{A^{(m)}}\rangle \langle j| = \frac{1}{||A^{(m)}||} \sum_i |\Lambda(i, :; m)\rangle \langle i| \langle j|,$$
where $s_{A^{(m)}}$ is a vector whose $i$-th entry is $\frac{||A(i, :; m)||}{||A^{(m)}||}$. This operator $U_{Q_m}$ can be implemented using the technique developed in [44] for preconditioned linear solvers. To be more specific, according to the algorithm 3, we can rewrite the state $|A^{(m)}\rangle$ in (8) as
$$|A^{(m)}\rangle = \frac{1}{||A^{(m)}||} \sum_i |\Lambda(i, :; m)\rangle \langle i| \langle i|.$$  \hfill (B.6)

Now we can perform QSVE on the matrix $\Lambda^{(m)}$ following similar procedures as in [31]. First, the factorization $\Lambda^{(m)} = \hat{P}_m \hat{Q}_m$ can be easily verified. Moreover, we can prove that $2\hat{P}_m \hat{P}_m - I_{N \times N}$ is reflection and it can be implemented through $U_{P_k}$. Actually,
$$2\hat{P}_m \hat{P}_m - I_{N \times N} = 2\sum_i |i\rangle \langle i| \Lambda(i, :; m)\rangle \langle i| \langle i| - I_{N \times N} = U_{P_k} [2\sum_i |i\rangle \langle i| - I_{N \times N}] U_{P_k}^*,$$
where $2\sum_i |i\rangle \langle i| - I_{N \times N}$ is a reflection. The similar result holds for $2\hat{Q}_m \hat{Q}_m - I_{N \times N}$.

Now denote
$$W_m = (2\hat{P}_m \hat{P}_m - I_{N \times N})(2\hat{Q}_m \hat{Q}_m - I_{N \times N}).$$  \hfill (B.7)

Let $\Lambda^{(m)} = \sum_{i=0}^{N-1} \lambda_i^{(m)} v_i^{(m)} v_i^{(m)*}$ be the singular value decomposition of $\Lambda^{(m)}$. We can prove that the subspace spanned by $\{Q_m v_i^{(m)}\}$ is invariant under the unitary transformation $W_m$:
$$W_m \hat{Q}_m v_i^{(m)} = \frac{2\lambda_i^{(m)}}{||\Lambda^{(m)}||} \hat{P}_m v_i^{(m)} - Q [v_i^{(m)}],$$
\[ W_m \hat{u}_m | \hat{t}^{(m)}_i \rangle = \left( \frac{4 \tilde{\delta}_m^2}{\| A^{(m)} \|_F^2} - 1 \right) \hat{u}_m | \hat{t}^{(m)}_i \rangle - \frac{2 \tilde{\delta}_m}{\| A^{(m)} \|_F} \hat{Q}_m | \hat{t}^{(m)}_i \rangle. \]

The matrix \( W_m \) can be calculated under an orthonormal basis using the Schmidt orthogonalization. It is a rotation in the subspace spanned by its eigenvectors \( | \hat{t}^{(m)}_i \rangle \) with correspondent eigenvalues \( e^{\pm i \hat{t}^{(m)}_i} \), where \( \hat{t}^{(m)}_i \) is the rotation angle satisfying \( \cos(\hat{t}^{(m)}_i / 2) = \frac{\| A^{(m)} \|_F}{\| A^{(m)} \|_F} \), i.e.

\[
\hat{Q}_m | \hat{t}^{(m)}_i \rangle = \sqrt{2} (| \omega^{(m)}_i \rangle + | \omega^{(m)}_{-i} \rangle),
\]

\[
\hat{P}_m | \hat{t}^{(m)}_i \rangle = \sqrt{2} (e^{i \hat{t}^{(m)}_i / 2} | \omega^{(m)}_i \rangle + e^{-i \hat{t}^{(m)}_i / 2} | \omega^{(m)}_{-i} \rangle).
\]

Here, we choose the input state as the Kronecker product form of the normalized matrix \( \hat{\lambda}^{(m)} \) represented in the svd, i.e. \( \hat{\lambda}^{(m)} = \frac{1}{\| A^{(m)} \|_F} \sum_{i} \hat{\delta}_i^{(m)} | \hat{t}^{(m)}_i \rangle \langle | \hat{t}^{(m)}_i \rangle \). Then

\[
I_{N_i} \otimes U_{\hat{Q}_m} \hat{\lambda}^{(m)} = \frac{1}{\| A^{(m)} \|_F} \sum_{i} \sqrt{2} \hat{\delta}_i^{(m)} | \hat{t}^{(m)}_i \rangle \langle | \hat{t}^{(m)}_i \rangle (| \omega^{(m)}_i \rangle + | \omega^{(m)}_{-i} \rangle).
\]

(B.7)

Performing the phase estimation on \( W_m \) with running time \( O(N \text{polylog} N / \epsilon_{\text{SVE}}^2) \) for \( N_1 = N_2 = N_3 = N \), and computing the estimated singular value of \( A^{(m)} \) through oracle \( \hat{\delta}_i^{(m)} = \| A^{(m)} \|_F \cos(\hat{t}^{(m)}_i / 2) / \sqrt{2} \), we obtain

\[
\frac{1}{\| A^{(m)} \|_F} \sum_{i} \sqrt{2} \hat{\delta}_i^{(m)} | \hat{t}^{(m)}_i \rangle \langle | \hat{t}^{(m)}_i \rangle (| \omega^{(m)}_i \rangle | \hat{P}^{(m)}_i \rangle + | \omega^{(m)}_{-i} \rangle | - \hat{P}^{(m)}_i \rangle | \pi_1^{(m)} \rangle).
\]

(B.8)

we next uncompute the phase estimation procedure and then apply the inverse of \( I_{N_i} \otimes U_{\hat{Q}_m} \) to obtain the desired state (12) in theorem 2.

\[ \square \]

**ORCID iDs**

Guofeng Zhang @ https://orcid.org/0000-0001-5854-5247

**References**

[1] Kilmer M E and Martin C D 2011 Factorization strategies for third-order tensors Linear Algebr. Appl. 435 641–58
[2] Zhang Z and Aaron S 2016 Exact tensor completion using t-svd IEEE Trans. Signal Process. 65 1511–26
[3] Zhang Z, Ely G, Aaron S, Hao N and Kilmer M 2014 Novel methods for multilinear data completion and de-noising based on tensor-svd Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 3842–9
[4] Ely G, Aaron S, Hao N and Kilmer M E 2015 3d seismic data completion and denoising using a novel class of tensor decompositions Geophysics 80 V83–95
[5] Zhou P, Lu C, Lin Z and Zhang C 2017 Tensor factorization for low-rank tensor completion IEEE Trans. Image Process. 27 1152–63
[6] Kolda T G and Bader B W 2009 Tensor decompositions and applications SIAM Rev. 51 455–500
[7] Qi L and Luo Z 2017 Tensor Analysis: Spectral Theory and Special Tensors (Philadelphia: SIAM)
[8] Qi L, Chen H and Chen Y 2018 Tensor Eigenvalues and Their Applications 39 (Berlin: Springer)
[9] Rendle S and Schmidt-Thieme L 2010 Pairwise interaction tensor factorization for personalized tag recommendation Proceedings of the 3rd ACM International Conference on Web Search and Data Mining pp 81–90
[10] Rendle S, Balgy Marinho L, Nanopoulos A and Schmidt-Thieme L 2009 Learning optimal ranking with tensor factorization for tag recommendation Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining pp 727–36
[11] Hu S, Qi L and Zhang G 2016 Computing the geometric measure of entanglement of multipartite pure states by means of non-negative tensors Phys. Rev. A 93 012304
[12] Zhang G 2017 Dynamical analysis of quantum linear systems driven by multi-channel multi-photon states Automatika 83 186–98
[13] Qi L, Zhang G, Braun D, Bohnet-Waldraff F and Giraud O 2017 Regularly decomposable tensors and classical spin states Communications in Mathematical Sciences 15 1651–1665
[14] Qi L, Zhang G and Ni G 2018 How entangled can a multi-party system possibly be? Phys. Lett. A 382 1465–71
[15] Rebentrost P, Schuld M, Wossnig L, Petruccione F and Lloyd S 2019 Quantum gradient descent and Newton’s method for constrained polynomial optimization New J. Phys. 21 073023
[16] Zhang M, Ni G and Zhang G 2020 Iterative methods for computing u-eigenvalues of non-symmetric complex tensors with application in quantum entanglement Comput. Optim. Appl. 75 779–98
[17] Wu M, He S, Zhang Y, Chen J, Sun Y, Liu Y-Y, Zhang J and Poor H V 2019 A tensor-based framework for studying eigenvector multicentrality in multilayer networks Proc. Natl Acad. Sci. 116 15407–13
[18] Huggins W, Patil P, Mitchell B, Whaley K B and Stoudenmire E M 2019 Towards quantum machine learning with tensor networks Quantum Science and Technology 4 024001
[19] Ma Y, Wang Y and Tresp V 2020 Quantum machine learning algorithm for knowledge graphs arXiv:2001.01077
[20] Cenon P 2002 Tensor decompositions Mathematics in Signal Processing V 1–24
[21] Oseledets I V 2011 Tensor-train decomposition SIAM J. Sci. Comput. 33 2295–317
[22] Tucker L R 1966 Some mathematical notes on three-mode factor analysis Psychometrika 31 279–311
[23] De Lathauwer L, De Moor B and Vandewalle J 2000 A multilinear singular value decomposition SIAM J. Matrix Anal. Appl. 21 1253–78
[24] Gu L, Wang X and Zhang G 2019 Quantum higher order singular value decomposition 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC) pp 1166–71
[25] Miao Y, Qi L and Wei Y 2020 Generalized tensor function via the tensor singular value decomposition based on the t-product Linear Algebra. Appl. 590 258–303

[26] Martin C D, Shafer R and LaRue B 2013 An order-p tensor factorization with applications in imaging SIAM J. Sci. Comput. 35 A474–90

[27] Hao N, Kilmer M E, Braman K and Hoover R C 2013 Facial recognition using tensor-tensor decompositions SIAM J. Imag. Sci. 6 437–63

[28] Song G, Ng M K and Zhang X 2020 Robust tensor completion using transformed tensor singular value decomposition Numerical Linear Algebra with Applications 27 e2299

[29] Zhang I, Saibaba A K, Kilmer M E and Aeron S 2018 A randomized tensor singular value decomposition based on the t-product Numerical Linear Algebra with Applications 25 e2179

[30] Nielsen M A and Chuang I 2010 Quantum Computation and Information (London: Cambridge University Press)

[31] Kerenidis I and Prakash A 2017 Quantum recommendation systems 8th Innovations in Theoretical Computer Science Conference (ITCS 2017), vol 67, Leibniz International Proceedings in Informatics (LIPIcs) (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany) ed C H Papadimitriou pp 49:1–49:21

[32] Symeonidis P, Nanopoulos A and Manolopoulos Y 2008 Tag recommendations based on tensor dimensionality reduction Proceedings of the 2008 ACM conference on Recommender systems 43–50

[33] Rebentrost P, Steffens A, Marvian I and Lloyd S 2018 Quantum singular-value decomposition of nonsparse low-rank matrices Phys. Rev. A 97 012327

[34] Rebentrost P, Steffens A, Marvian I and Lloyd S 2018 Quantum singular-value decomposition of nonsparse low-rank matrices Phys. Rev. A 97 012327

[35] Maroulis S, Boutis I and Kalogeraki V 2016 Context-aware point of interest recommendation using tensor factorization 2016 IEEE International Conference on Big Data (Big Data) (IEEE) pp 963–8

[36] Liao J, Tang J, Zhao X and Shang H 2018 Improving poi recommendation via dynamic tensor completion Sci. Program. 2018 2018

[37] Song G, Ng M K and Zhang X 2020 Robust tensor completion using transformed tensor singular value decomposition Numerical Linear Algebra with Applications 27 e2299

[38] Teixeira M and Rodriguez D 1995 A class of fast cyclic convolution algorithms based on block pseudocirculants IEEE Signal Process Lett. 2 92–4

[39] Clader B D, Jacobs B C and Sprouse C R 2013 Preconditioned quantum linear system algorithm Phys. Rev. Lett. 110 250504

[40] Shao C 2018 From linear combination of quantum states to grover’s searching algorithm arXiv:1807.09693

[41] Long G L 2011 Duality quantum computing and duality quantum information processing Int. J. Theor. Phys. 50 1303–18

[42] Shao C, Li Y and Li H 2019 Quantum algorithm design: techniques and applications Journal of Systems Science and Complexity 32 375–452

[43] Long G L 2006 General quantum interference principle and duality computer Commun. Theor. Phys. 45 825

[44] Shao C and Xiang H 2018 Quantum circulant preconditioner for a linear system of equations Phys. Rev. A 98 062321