Quantum context-aware recommendation systems based on 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.

Abstract

In this paper, we propose a quantum algorithm for recommendation systems which incorporates the contextual information of users to the personalized recommendation. The preference information of users is encoded in a third-order tensor of dimension N which can be approximated by the truncated tensor singular value decomposition (t-svd) of the subsample tensor. Unlike the classical algorithm that reconstructs the approximated preference tensor using truncated t-svd, our quantum algorithm obtains the recommended product under certain context by measuring the output quantum state corresponding to an approximation of a user's dynamic preferences. The algorithm achieves the time complexity $\mathcal{O}(\sqrt{k}N\text{polylog}(N))$, compared to the classical counterpart with complexity $\mathcal{O}(kN^3)$, where k is the truncated tubal-rank.

Keywords: context-aware recommendation systems, t-svd; quantum singular value estimation; quantum Fourier transform

1 Introduction

Machine learning is a branch of artificial intelligence and is increasingly ubiquitous in various areas such as natural language processing, data mining, biological analysis, etc. However, a major deficiency of many machine learning algorithms is their high computational and storage cost when processing big data. On the other hand, quantum computer is considered as one of the most promising and emerging technologies, and its development has made great progress in recent years. Considering high demanding computational power of machine learning and the fast development of quantum technology, researchers are developing a new interdisciplinary research field, quantum machine learning.

Quantum machine learning explores the interaction between quantum computing and machine learning, by investigating how quantum techniques, e.g., superposition and entanglement, can be used to speed up some classical machine learning problems. Successful examples are quantum support vector machine (QSVM) [36], quantum principle component analysis (QPCA) [22], among others. In most cases, quantum computing is supposed to deal with quantum data, as commented in [3]. Hence, the classical data should be preprocessed into quantum data using some methods like QRAM [10, 18] so that quantum algorithms can proceed as desired. As quantum features such as parallelism and entanglement can be used to accelerate some computational procedures which classical operations are generally regarded as inefficient, it is reasonable to assume that the performance of quantum computers outperforms classical computers on certain machine learning problems.

Tensor refers to a multi-dimensional array of numbers, thus it can represent more complex structures of higher-order data. Applications involving tensors include image deblurring, video recovery, denoising, data completion, multi-partite quantum systems, networks and machine learning [19, 58, 59, 9, 60, 20, 32, 31, 40, 39, 14, 55, 33, 34, 37, 57, 53, 15, 25], due to the flexibility of tensors in representing data. Some of these applications make use of various tensor decompositions including CANDECOMP/PARAFAC (CP) [7], TUCKER [51], higher-order singular value decomposition (HOSVD) [8, 55, 11], tensor-train decomposition (TT) [29], and tensor singular value de-

[*] Corresponding author: guofeng.zhang@polyu.edu.hk

composition (t-svd) [19, 59, 25]. A review of the applications of tensor networks in quantum physics can be found in [28].

Plenty of research has been carried out on t-svd in the last decade. The concept of t-svd was first proposed by Kilmer and Martin [19] for third-order tensors. Later, Martin et al. [26] extended it to higher-order tensors. The t-svd algorithm is superior to TUCKER and CP decompositions in the sense that it extends the familiar matrix svd strategy to tensors, thus avoiding the loss of information inherent in flattening tensors used in TUCKER and CP decompositions. Compared with HOSVD, t-svd also has optimality properties similar to the truncated svd for matrices, hence t-svd is shown to have better performance than HOSVD when applied to facial recognition [12], tensor completion [58, 45]. Another advantage of t-svd is that it can be obtained by computing matrix svd in the Fourier domain; the similar idea allows other matrix factorization techniques like QR decomposition to be extended to tensors easily.

It is well known that truncating k-term matrix svd provides the best rank-k approximation of a matrix in both ℓ_2 norm and Frobenius norm. This raises a question whether the truncated tensor decompositions also have the similar optimality. The CP decompostion expresses a tensor as a sum of outer products of vectors (rank-1 tensor), and CP-rank is defined as the minimal number of rank-1 tensor necessary to construct the tensor, but calculating the CP-rank and the rank-k CP approximation are numerically unstable. Also, truncated Tucker decomposition does not yield the best fit of the original tensor [56]. In contrast, the t-svd gives an optimal approximation of a tensor in Frobenius norm [56]. Unfortunately, the cost of computing t-svd factorization is prohibitively expensive especially for very high dimensional tensors, e.g., the cost is $\mathcal{O}(N^4 + N^3\text{log}N)$ for a third-order tensor with dimension N. Therefore, much work focuses on computing low-rank tensor approximations based on t-svd with comparatively low cost. For example, randomized tensor low-rank representations based on the t-svd can give the nearly optimal approximation with complexity $\mathcal{O}(kN^3 + N^3\text{log}N)$ for a third-order tensor with dimension N [56].

The quantum algorithm that we propose in this work implements a machine learning task, namely, context-aware recommendation systems in which preference information is encoded in a third-order tensor. For recommendation systems modeled by an $m \times n$ preference matrix, Kerenidis and Prakash designed a quantum algorithm that offers recommendations by just sampling from an approximated preference matrix [18]. Therefore, the running time is only $\mathcal{O}(\text{poly}(k)\text{polylog}(mn))$ if the preference matrix has a good rank-k approximation. To achieve this, they projected a state corresponding to a user's preferences to the approximated row space spanned by singular vectors whose corresponding singular values are greater than the prescribed threshold. After measuring this projected state in a computational basis, they got recommended product indices for the input user.

In recommendation systems algorithms, most model-based Collaborative Filtering approaches, e.g. matrix factorization, fail to model context information [16]. Context is an important factor to consider in personalized recommendation systems. In [54], it is demonstrated by an experiment that when a recommendation system is modeled by a tensor whose third dimension is context (e.g. time), accuracy could be improved compared to the non-contextual modeling. Hence, most recent research has focused on developing context-aware recommendation systems modeled by tensors; see, e.g., [46, 59, 35, 16]. The classical third-order tensor recommendation systems algorithms based on tensor factorizations, such as the truncated t-svd [59] and the truncated HOSVD (T-HOSVD) [46], all have the computational complexity at least $\mathcal{O}(kN^3)$.

Taking into account the effectiveness of third-order tensor modeling and high cost of the truncated t-svd algorithm, in this paper we extend Kerenidis and Prakash's algorithm [18] from matrices to third-order tensors and propose a quantum context-aware recommendation systems algorithm based on truncated t-svd factorization. In general, a user's preference in a certain context is very likely to affect the recommendation for him/her at other contexts. As the quantum Fourier transform (QFT) used in our t-svd algorithm is performed to bind a user's preferences in different context together, our quantum algorithm is well suited to context-aware recommendation systems. The general idea of our algorithm is to approximate the observed preference tensor using truncated matrix svd in the Fourier domain. Similar to the low-rank assumption adopted in quantum 2D recommendation systems [18], we exploit low structural and informational complexity of the data, expressed as low tubal-rank hypothesis of the underlying data, by which it is feasible to predict the missing entries. Moreover, taking advantage of the quantum parallelism, the quantum singular value estimation of all frontal slices after QFT can be performed parallelly, which dramatically reduces the time complexity to $\mathcal{O}(N\sqrt{k}\text{polylog}(N))$.

In Section 4, our numerical experiments show that the truncated t-svd is better than T-HOSVD and TT decompositions when applied to context-aware recommendation systems. In fact, our quantum algorithm suits the context-aware recommendation systems model very well since it is not necessary to reconstruct the entire tensor. All we need is to recommend products that a user prefers, which corresponds to measuring certain times a quantum state representing a user' approximated preference, so our quantum algorithm can provide good recommendations with much lower complexity.

The rest of this paper is organized as follows. Preliminaries are given in Section 2. In Section 3, we propose our main algorithm, quantum context-aware recommendation systems algorithm. In Section 4, we numerically validate the classical counterpart of our model with real datasets. At last, we compare Tang's 2D quantum-inspired recommendation systems with our algorithm in Section 5 and we conclude the paper in Section 6.

2 Preliminaries

We first introduce some preliminary material on tensors and notations in Section 2.1. In Section 2.2, we review the definition of t-product and the classical (namely, non-quantum) t-svd algorithm proposed by Kilmer et al. [19] in 2011. Then in Section 2.3, we briefly review the quantum singular value estimation algorithm (QSVE) proposed by Kerenidis and Prakash [18].

2.1 Tensor background and notation

An order-p tensor $\mathcal{A}$ can be represented as a multi-dimensional array of data, i.e.,

$$\mathcal{A} = (a_{i_1 i_2 \cdots i_p}) \in \mathbb{C}^{N_1 \times N_2 \times \cdots \times N_p}.$$

The order of a tensor is the number of modes; for example, $\mathcal{A} \in \mathbb{C}^{N_1 \times N_2 \times N_3}$ is a third-order tensor of complex numbers with dimension N_i for mode i, $i = 1, 2, 3$, respectively. In this way, a matrix A can be regarded as a second-order tensor, and a vector $\boldsymbol{x}$ is a first-order tensor. A third-order tensor can be imagined as a cube of data, and a slice of a tensor can be regarded as a matrix defined by fixing one index. We use terms frontal slice $\mathcal{A}(:,:,i)$, horizontal slice $\mathcal{A}(i,:,:)$ and lateral slice $\mathcal{A}(:,i,:)$ (see FIG.1) to specify which index in three modes is fixed. A tube of size $1 \times 1 \times N_3$ can be regarded as a vector and it is defined by fixing all indices but the last one, e.g., $\mathcal{A}(i,j,:)$ is the (i,j)-th tube of $\mathcal{A}$.

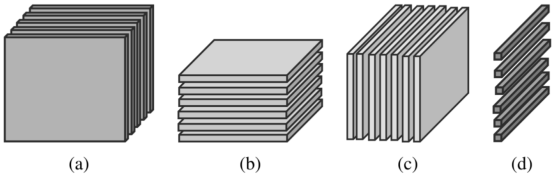

Figure 1: (a) frontal slices, (b) horizontal slices, (c) lateral slices of a third-order tensor. (d) A lateral slice as a vector of tubes.

Notation. In this paper, to facilitate the distinction between vectors, matrices, and higher-order tensors, we use different representations based on their types. 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 ($\boldsymbol{x}$, $\boldsymbol{y}$, $\cdots$). We use $A^{(i)}$ to denote the i-th frontal slice $\mathcal{A}(:,:,i)$ for short. DFT($\boldsymbol{u}$) refers to performing the discrete Fourier transform (DFT) on $\boldsymbol{u}$, which is computed by the fast Fourier transform represented in Matlab notation fft($\boldsymbol{u}$). The tensor after DFT along the third mode of $\mathcal{A}$ is denoted by $\hat{\mathcal{A}}$, i.e., $\hat{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$. Hence we have $\mathcal{A} = \text{ifft}(\hat{\mathcal{A}}, [], 3)$. 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: $\boldsymbol{u} \circledast \boldsymbol{v}$ refers to the circular convolution between vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, $\odot$ is the Hadamard product and $\mathcal{A} * \mathcal{B}$ represents the t-product between tensors $\mathcal{A}$ and $\mathcal{B}$.

2.2 The t-svd algorithm and t-product

In this subsection, we provide the definition of a new type of multiplication between tensors, t-product, and give corresponding definitions of identical, transpose, and orthogonal tensors from [19]. Then we give a t-svd algorithm based on t-product.

Definition 1. ***circular convolution*** *[48]*
Let $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^N$. The circular convolution between $\boldsymbol{u}$ and $\boldsymbol{v}$ produces a vector $\boldsymbol{x}$ of the same size, defined as

$$\boldsymbol{x} \equiv \boldsymbol{u} \circledast \boldsymbol{v} \triangleq \mathrm{circ}(\boldsymbol{u})\boldsymbol{v},$$

where

$$\mathrm{circ}(\boldsymbol{u}) = \begin{bmatrix} u_0 & u_{N-1} & \cdots & u_1 \\ u_1 & u_0 & \cdots & u_2 \\ \vdots & \vdots & \ddots & \vdots \\ u_{N-1} & u_{N-2} & \cdots & u_0 \end{bmatrix}.$$

Theorem 1. ***Cyclic Convolution Theorem*** *[48]*
Given $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^N$, let $\boldsymbol{x} = \boldsymbol{u} \circledast \boldsymbol{v}$, and $\circledast$ refers to the circular convolution. We have

$$\mathrm{DFT}(\boldsymbol{x}) = \mathrm{DFT}(\boldsymbol{u}) \odot \mathrm{DFT}(\boldsymbol{v}), \tag{1}$$

where $\odot$ is the Hadamard product.

If a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is regarded as an $N_1 \times N_2$ matrix of tubes of dimension N_3, whose (i,j)-th entry (a tube) is $\mathcal{A}(i,j,:)$, then based on the definition of circular convolution between vectors, the t-product between tensors can be defined.

Definition 2. ***t-product*** *[19]*
Let $\mathcal{M} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{N} \in \mathbb{R}^{N_2 \times N_4 \times N_3}$. The t-product $\mathcal{M} * \mathcal{N}$ is an $N_1 \times N_4 \times N_3$ tensor, denoted by $\mathcal{A}$, whose (i,j)-th tube $\mathcal{A}(i,j,:)$ is the sum of the circular convolution between corresponding tubes in the i-th horizontal slice of the tensor $\mathcal{M}$ and the j-th lateral slice of the tensor $\mathcal{N}$, i.e.,

$$\mathcal{A}(i,j,:) = \sum_{k=0}^{N_2-1} \mathcal{M}(i,k,:) \circledast \mathcal{N}(k,j,:). \tag{2}$$

According to Theorem 1 and Definition 2, we have

$$\begin{aligned} &\mathrm{DFT}(\mathcal{A}(i,j,:)) \\ &= \sum_{k=0}^{N_2-1} \mathrm{DFT}(\mathcal{M}(i,k,:)) \odot \mathrm{DFT}(\mathcal{N}(k,j,:)), \end{aligned} \tag{3}$$

for $i = 0, \cdots, N_1 - 1$; $j = 0, \cdots, N_4 - 1$. Let $\hat{\mathcal{A}}$ be the tensor whose (i,j)-th tube is $\mathrm{DFT}(\mathcal{A}(i,j,:))$. Then equation (3) becomes $\hat{\mathcal{A}}(i,j,:) = \sum_{k=0}^{N_2-1} \hat{\mathcal{M}}(i,k,:) \odot \hat{\mathcal{N}}(k,j,:)$, which can also be written in the form $\hat{\mathcal{A}}(i,j,l) = \sum_{k=0}^{N_2-1} \hat{\mathcal{M}}(i,k,l)\hat{\mathcal{N}}(k,j,l)$ for the l-th frontal slices of these tensors. Therefore, $\hat{A}^{(l)} = \hat{M}^{(l)}\hat{N}^{(l)}$. The following theorem summarizes the idea stated above.

Theorem 2. *[19] For tensors $\mathcal{M} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{N} \in \mathbb{R}^{N_2 \times N_4 \times N_3}$, the equivalence relation*

$$\mathcal{A} = \mathcal{M} * \mathcal{N} \quad \Longleftrightarrow \quad \hat{A}^{(l)} = \hat{M}^{(l)}\hat{N}^{(l)} \tag{4}$$

holds for $l = 0, \cdots, N_3 - 1$. Moreover, for another tensor $\mathcal{T} \in \mathbb{R}^{N_4 \times N_5 \times N_3}$, we have

$$\mathcal{A} = \mathcal{M} * \mathcal{N} * \mathcal{T} \quad \Longleftrightarrow \quad \hat{A}^{(l)} = \hat{M}^{(l)}\hat{N}^{(l)}\hat{T}^{(l)}, \tag{5}$$

for $l = 0, \cdots, N_3 - 1$.

Definition 3. ***tensor Frobenius norm*** *[19]*
The Frobenius norm of a third-order tensor $\mathcal{A} = (a_{ijk})$ is defined as $||\mathcal{A}||_F = \sqrt{\sum_{i,j,k} |a_{ijk}|^2}$.

Definition 4. ***tensor transpose*** *[19]*
The transpose of a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, denoted as $\mathcal{A}^T$, is obtained by transposing all the frontal slices and then reversing the order of the transposed frontal slices 1 through $N_3 - 1$.

Definition 5. ***identity tensor*** *[19]*
The identity tensor $\mathcal{I} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ is a tensor whose first frontal slice $I^{(0)}$ is an $N_1 \times N_1$ identity matrix and all the other frontal slices are zero matrices.

Definition 6. ***orthogonal tensor*** *[19]*
A tensor $\mathcal{U} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ is an orthogonal tensor if it satisfies $\mathcal{U}^T * \mathcal{U} = \mathcal{U} * \mathcal{U}^T = \mathcal{I}$.

Theorem 3. ***tensor singular value decomposition (t-svd)*** *[19]*
For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, its t-svd is given by $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$, where $\mathcal{U} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ and $\mathcal{V} \in \mathbb{R}^{N_2 \times N_2 \times N_3}$ are orthogonal tensors, and every frontal slice of $\mathcal{S} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is a diagonal matrix (FIG.2).

Algorithm 1 t-svd for third-order tensors [19]

```
**Input**: $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$
**Output**: $\mathcal{U} \in \mathbb{R}^{N_1 \times N_1 \times N_3}, \mathcal{S} \in \mathbb{R}^{N_1 \times N_2 \times N_3}, \mathcal{V} \in \mathbb{R}^{N_2 \times N_2 \times N_3}$
$\hat{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$;
**for** $i = 0, \cdots, N_3 - 1$ **do**
    $[U, S, V] = \text{svd}(\hat{\mathcal{A}}(:,:,i))$;
    $\hat{\mathcal{U}}(:,:,i) = U; \hat{\mathcal{S}}(:,:,i) = S; \hat{\mathcal{V}}(:,:,i) = V$;
**end for**
$\mathcal{U} = \text{ifft}(\hat{\mathcal{U}}, [], 3); \mathcal{S} = \text{ifft}(\hat{\mathcal{S}}, [], 3); \mathcal{V} = \text{ifft}(\hat{\mathcal{V}}, [], 3)$.
```

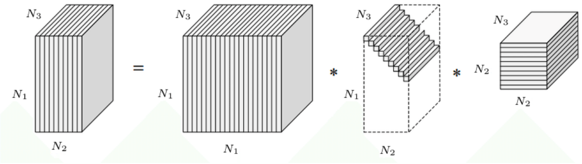

Figure 2: The t-svd of $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$.

Definition 7. ***tubal-rank*** *[19]* *The tubal-rank of a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is the number of nonzero tubes $\mathcal{S}(i,i,:)$, $i = 0, \cdots, \min(N_1, N_2) - 1$, in t-svd factorization.*

Remark 1. *In the t-svd literature, the diagonal elements of the tensor $\mathcal{S}$ are called the singular values of $\mathcal{A}$. Moreover, the l_2 norms of the nonzero tubes $\mathcal{S}(i,i,:)$ are in descending order, i.e.,*

$$||\mathcal{S}(0,0,:)||_2 \geq ||\mathcal{S}(1,1,:)||_2 \geq \cdots$$
$$\geq ||\mathcal{S}(\min(N_1,N_2)-1, \min(N_1,N_2)-1,:)||_2.$$

However, it should be noticed that the diagonal elements of $\mathcal{S}$ may be unordered and even negative due to the inverse DFT. As a result, when doing tensor truncation in Section 3 to get quantum recommendation systems, we use $\hat{\mathcal{S}}$ instead of $\mathcal{S}$ as the diagonal elements of the former are non-negative and ordered in descending order.

Definition 8. ***multi-rank*** *[19]* *The multi-rank of a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is a vector in $\mathbb{R}^{N_3}$ whose i-th entry equals to the rank of $\hat{\mathcal{A}}(:,:,i)$.*

Many important applications of the t-svd algorithm, such as data compression and completion, utilize the optimality of truncated t-svd in the sense that it gives an optimal approximation of a tensor measured by the Frobenius norm. The following theorem describes this property, which is the theoretical basis of our quantum algorithm for recommendation systems to be developed in Sections 3.2.

Lemma 1. *[19, 56] Suppose the t-svd of the tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$. Then we have*

$$\mathcal{A} = \sum_{i=0}^{\min(N_1,N_2)-1} \mathcal{U}(:,i,:) * \mathcal{S}(i,i,:) * \mathcal{V}(:,i,:)^T,$$

where the matrices $\mathcal{U}(:,i,:)$ and $\mathcal{V}(:,i,:)$ and the vector $\mathcal{S}(i,i,:)$ are regarded as tensors of order 3. For $1 \leq k < \min(N_1, N_2)$, define $\mathcal{A}_k \triangleq \sum_{i=0}^{k-1} \mathcal{U}(:,i,:) * \mathcal{S}(i,i,:) * \mathcal{V}(:,i,:)^T$. Then

$$\mathcal{A}_k = \arg \min_{\tilde{\mathcal{A}} \in \mathcal{M}_k} ||\mathcal{A} - \tilde{\mathcal{A}}||_F,$$

where $\mathcal{M}_k = \{\mathcal{X} * \mathcal{Y} | \mathcal{X} \in \mathbb{R}^{N_1 \times k \times N_3}, \mathcal{Y} \in \mathbb{R}^{k \times N_2 \times N_3}\}$. Therefore, $||\mathcal{A} - \mathcal{A}_k||_F$ is the theoretical minimal error, given by $||\mathcal{A} - \mathcal{A}_k||_F = \sqrt{\sum_{i=k}^{\min(N_1,N_2)-1} ||\mathcal{S}(i,i,:)||_2^2}$.

2.3 Quantum singular value estimation

Kerenidis and Prakash [18] proposed a quantum algorithm to estimate the singular values of a matrix, named by the quantum singular value estimation (QSVE). With the introduction of a data structure, see Lemma 2 below, in which the rows of the matrix are stored, the QSVE algorithm can prepare the quantum states corresponding to the rows of the matrix efficiently.

Lemma 2. *[18, Theorem 5.1] Consider a matrix $A \in \mathbb{R}^{N_1 \times N_2}$ with ι nonzero entries. Let A_i be its i-th row, and $\boldsymbol{s}_A = \frac{1}{||A||_F} \left[||A_0||_2, ||A_1||_2, \cdots, ||A_{N_1-1}||_2\right]^T$. There exists a data structure storing the matrix A in $\mathcal{O}(\iota \log^2(N_1 N_2))$ space such that a quantum algorithm having access to this data structure can perform the mapping $U_P : |i\rangle|0\rangle \rightarrow |i\rangle|A_i\rangle$, for $i = 0, \cdots, N_1 - 1$ and $U_Q : |0\rangle|j\rangle \rightarrow |\boldsymbol{s}_A\rangle|j\rangle$, for $j = 0, \cdots, N_2 - 1$ in time* polylog$(N_1 N_2)$.

The explicit description of the QSVE is given in [18] and the following lemma summarizes the main ideas.

Lemma 3. *[18, Theorem 5.2] Let $A \in \mathbb{R}^{N_1 \times N_2}$ and $\boldsymbol{x} \in \mathbb{R}^{N_2}$ be stored in the data structure as mentioned* *in Lemma 2. Let the singular value decomposition of A be $A = \sum_{l=0}^{r-1} \sigma_l \left|u_l\right\rangle \left\langle v_l\right|$, where $r = \min(N_1, N_2)$. The input state $|x\rangle$ can be represented in the eigenstates of A, i.e. $|x\rangle = \sum_{l=0}^{N_2-1} \beta_l \left|v_l\right\rangle$. Let $\epsilon > 0$ be the precision parameter. Then there is a quantum algorithm, denoted as U_{SVE}, that runs in time $\mathcal{O}(\mathrm{polylog}(N_1 N_2)/\epsilon)$ and achieves*

$$U_{\mathrm{SVE}}\left(|x\rangle\,|0\rangle\right) = \sum_{l=0}^{N_2-1} \beta_l \left|v_l\right\rangle \left|\overline{\sigma}_l\right\rangle$$

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

3 Quantum algorithm for recommendation systems modeled by third-order tensors

In this section, we will first introduce the notation adopted in this section and then give an overview of Algorithm 2. In Section 3.1, the main ideas and assumptions of the algorithm are summarized. In Section 3.2, we explain each step of our algorithm in detail, followed by a summary in Algorithm 2. Error analysis is given in Section 3.3 and complexity analysis is conducted in Section 3.4.

The preference information of users is stored in a third-order tensor $\mathcal{A} \in \mathbb{R}^{N \times N \times N}$, called the preference tensor, whose three modes represent user(i), product(j) and context(t) respectively. The tube $\mathcal{A}(i, j, :)$ is regarded as the rating scores of the user i for the product j under different contexts. For user i in context t, the entry $\mathcal{A}(i, j, t)$ takes value 1 indicating the product j is "good" and value 0 otherwise. In this sense, a triplet (i, j, t) is called a good recommendation if $\mathcal{A}(i, j, t) = 1$ or a bad recommendation otherwise. Let tensor $\mathcal{T}$ be the random tensor obtained by sampling from the tensor $\mathcal{A}$ with probability p and $\hat{\mathcal{T}}$ be the tensor obtained by performing the QFT along the third mode of $\mathcal{T}$. We use $\hat{T}^{(m)}$ to denote the m-th frontal slice of tensor $\hat{\mathcal{T}}$. $\hat{T}^{(m)}_{\geq \tau_m}$ is formed by projecting $\hat{T}^{(m)}$ onto the space spanned by the singular vectors whose corresponding singular values are greater than the threshold τ_m. $\hat{\mathcal{T}}_{\geq \boldsymbol{\tau}}$ denotes the tensor whose m-th frontal slice is $\hat{T}^{(m)}_{\geq \tau_m}$. Here, the threshold $\boldsymbol{\tau}$ of tensor $\hat{\mathcal{T}}$ actually denotes a list of thresholds $\{\tau_0, \cdots, \tau_{N-1}\}$ since different frontal slices $\hat{T}^{(m)}$ have their corresponding thresholds τ_m. $\mathcal{T}_{\geq \boldsymbol{\tau}}$ denotes the tensor obtained by performing the inverse QFT along the third mode of $\hat{\mathcal{T}}_{\geq \boldsymbol{\tau}}$.

3.1 Main ideas

We will propose Algorithm 2 which recommends a product j to a user i at a certain context t_0. The algorithm is inspired by the matrix recommendation systems method [1, 18] and a tensor reconstruction algorithm [59]. The main ideas are summarized in the following flow chart:

$$\mathcal{T}(i,:,:) \xrightarrow{\text{QFT}} \hat{\mathcal{T}}(i,:,:) \xrightarrow{\text{tube}} \hat{\mathcal{T}}(i,:,m) \xrightarrow{\text{approximation}}$$
$$\hat{\mathcal{T}}_{\geq \boldsymbol{\tau}}(i,:,m) \xrightarrow{\text{stack up}} \hat{\mathcal{T}}_{\geq \boldsymbol{\tau}}(i,:,:) \xrightarrow{\text{iQFT}} \mathcal{T}_{\geq \boldsymbol{\tau}}(i,:,:).$$

Suppose there is a hidden preference tensor $\mathcal{A}$ which is assumed to have a low tubal-rank k. This low tubal-rank assumption is also adopted in the classical truncated t-svd data completion problem [58, 45]. In practical applications, only a part of the entries of $\mathcal{A}$ can be observed. Our goal is to predict the missing entries and recommend product which has a high predicted value. This partially observed tensor is called the subsample tensor $\mathcal{T}$ which is sparse in general. Here, we use the subsampling method proposed in [1] to get $\mathcal{T}$, which is also adopted in [18]. Specifically, $\mathcal{T}_{ijt} = \mathcal{A}_{ijt}/p$ with probability p and $\mathcal{T}_{ijt} = 0$ otherwise, where p is called the subsampling probability. Clearly, the expectation $\mathrm{E}(\mathcal{T}) = \mathcal{A}$. In this sense, our algorithm can be understood as an approximate reconstruction of the hidden preference tensor $\mathcal{A}$, of which we are only given a sample ratings in the form of the sparse tensor $\mathcal{T}$.

Our algorithm recommends products according to the low tubal-rank tensor $\mathcal{T}_{\geq \boldsymbol{\tau}}$ which can be proved to an approximation of the hidden preference tensor $\mathcal{A}$. Inspired by the t-svd, this approximation process is conducted under the Fourier domain, as shown in the above flow chart. Next we explain each step of the flow chart in detail. Given a state $|\mathcal{T}(i,:,:)\rangle$ representing the observed preference information of user i, we first perform QFT on its last register, obtaining the state $|\hat{\mathcal{T}}(i,:,:)\rangle$. For each $\hat{T}^{(m)}$, which is the m-th frontal slice of the tensor $\hat{\mathcal{T}}$, we perform the modified QSVE on this matrix using the input state $|\hat{\mathcal{T}}(i,:,m)\rangle$ and truncate the resulting singular values with threshold τ_m, $m = 0, \cdots, N_3 - 1$

obtaining the state $|\hat{\mathcal{T}}_{\geq\boldsymbol{\tau}}(i,:,m)\rangle$. Stacking tubes $\hat{\mathcal{T}}_{\geq\boldsymbol{\tau}}(i,:,m)$ $(m=0,\ldots,N-1)$ yields the horizontal slice $\hat{\mathcal{T}}_{\geq\boldsymbol{\tau}}(i,:,:)$ which can be regarded as an approximation of $\hat{\mathcal{T}}(i,:,:)$. After the inverse QFT on $\hat{\mathcal{T}}_{\geq\boldsymbol{\tau}}(i,:,:)$, the horizontal slice $\mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)$ is obtained. We prove in Section 3.3 that $\mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)$ is an approximation of $\mathcal{T}(i,:,:)$, hence it can approximate user i's preference $\mathcal{A}(i,:,:)$.

The approximation tensor $\mathcal{T}_{\geq\boldsymbol{\tau}}$ is non-sparse in general, thus we can get a recommended index based on the non-zero entries of $\mathcal{T}_{\geq\boldsymbol{\tau}}$. For a given context t, our algorithm provides a recommended product index j for a user i by just measuring the output quantum state $|\mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)\rangle$ in the computational basis. In Theorem 5, we provide an upper bound on the probability of the pair (i,j,t) being a bad recommendation. This upper bound could be small by taking reasonable values for the related parameters.

Assumption 1. *The following assumptions are used in Algorithm 2.*

1. *Every frontal slice of the subsample tensor $\mathcal{T}\in\mathbb{R}^{N\times N\times N}$ is stored in the data structure as mentioned in Lemma 2.*

2. *For all $i,m=0,\cdots,N-1$, we assume the tubes $\mathcal{A}(i,:,m)$ satisfy*
$$\frac{1}{1+\gamma}\frac{||\mathcal{A}||_F^2}{N^2}\leq||\mathcal{A}(i,:,m)||_2^2\leq(1+\gamma)\frac{||\mathcal{A}||_F^2}{N^2} \tag{6}$$
for a given $\gamma>0$.

The second assumption indicates that users are all typical users, that is, the number of preferred products of users is close to the average in any context m. These assumptions are also adopted in [18] for matrices, where they explain the rationality of these assumptions.

3.2 The algorithm

Given the hidden preference tensor $\mathcal{A}$, the sampling probability p , the assumed low tubal-rank k, and the precision $\epsilon_{\text{SVE}}^{(m)}$ of the modified QSVE on each $\hat{T}^{(m)}$, Algorithm 2 outputs the state corresponding to the approximation of the i-th horizontal slice $\mathcal{A}(i,:,:)$. The quantum circuits of Algorithm 2 is shown in FIG. 3. In what follows we explain the steps of this algorithm.

The dynamic preference tensor $\mathcal{A}\in\mathbb{R}^{N\times N\times N}$ can be interpreted as the preference matrix $\mathcal{A}(:,:,t)$ evolving over the context t. It is reasonable to believe that the tubes $\mathcal{A}(i,:,0),\cdots,\mathcal{A}(i,:,N-1)$ are related to each other because the preference of the same user i in different contexts is mutually influenced. Considering these relations, we merge tubes in the same horizontal slice together through the QFT after getting the subsample tensor $\mathcal{T}$. In other words, in **Step 1**, the QFT is performed on the last register of the input state
$$|\mathcal{T}(i,:,:)\rangle=\frac{1}{||\mathcal{T}(i,:,:)||_F}\sum_{j,t=0}^{N-1}\mathcal{T}(i,j,t)\,|j\rangle^d\,|t\rangle^e \tag{7}$$
to get
$$\begin{aligned}&|\hat{\mathcal{T}}(i,:,:)\rangle\\&=\frac{1}{||\mathcal{T}(i,:,:)||_F}\sum_{m=0}^{N-1}||\hat{\mathcal{T}}(i,:,m)||_2\,|\hat{\mathcal{T}}(i,:,m)\rangle^d\,|m\rangle^e,\end{aligned} \tag{8}$$
where $\omega=e^{2\pi\mathrm{i}/N}$ and
$$\begin{aligned}&|\hat{\mathcal{T}}(i,:,m)\rangle\\&=\frac{1}{\sqrt{N}||\hat{\mathcal{T}}(i,:,m)||_2}\sum_{j,t=0}^{N-1}\omega^{tm}\mathcal{T}(i,j,t)\,|j\rangle.\end{aligned} \tag{9}$$
Note that $||\hat{\mathcal{T}}(i,:,:)||_F=||\mathcal{T}(i,:,:)||_F$, since the Frobenius norm does not change under the Fourier transform. In FIG. 3, the input state $|\mathcal{T}(i,:,:)\rangle$ is represented in lines d and e with $\lceil\log N\rceil$ qubits, and the QFT is denoted as F in line e. The quantum cost and circuit for QFT are given in Appendix E.1.

In **Step 2**, a controlled operator
$$U=\sum_{m=0}^{N-1}U_{\text{SVE}}^{(m)}\otimes|m\rangle\langle m|^e \tag{10}$$
is performed on the state $|\hat{\mathcal{T}}(i,:,:)\rangle$, where $U_{\text{SVE}}^{(m)}$ denotes the modified quantum singular value estimation process on the matrix $\hat{T}^{(m)}$ with the input $|\hat{\mathcal{T}}(i,:,m)\rangle$. As $U_{\text{SVE}}^{(m)}$ follows similar steps of QSVE, its circuit is also similar to that of QSVE given in FIG. 8. The detailed process of $U_{\text{SVE}}^{(m)}$ is described in Theorem 4 and Appendix A. The quantum cost of implementing the operator U is analyzed in Appendix E.3. Because of the quantum parallelism,

the operator U performed on the superposition state $|\hat{\mathcal{T}}(i,:,:)\rangle$ is thus equivalent to $U_{\text{SVE}}^{(m)}$ performed on each of the components $|\hat{\mathcal{T}}(i,:,m)\rangle$ as a single input, i.e.,

$$U\,|\hat{\mathcal{T}}(i,:,:)\rangle = \frac{1}{||\mathcal{T}(i,:,:)||_F} \sum_{m=0}^{N-1} ||\hat{\mathcal{T}}(i,:,m)||_2 \left(U_{\text{SVE}}^{(m)}\,|\hat{\mathcal{T}}(i,:,m)\rangle^d\right)|m\rangle^e . \tag{11}$$

Next, we focus on $U_{\text{SVE}}^{(m)}\,|\hat{\mathcal{T}}(i,:,m)\rangle^d$ in (11). Suppose $\sum_{j=0}^{N-1} \hat{\sigma}_j^{(m)}\hat{u}_j^{(m)}\hat{v}_j^{(m)\dagger}$ is the svd of $\hat{T}^{(m)}$, we first express $|\hat{\mathcal{T}}(i,:,m)\rangle$ under the basis of $\hat{v}_j^{(m)}, j = 0,\cdots,N-1$, i.e.,

$$|\hat{\mathcal{T}}(i,:,m)\rangle = \sum_{j=0}^{N-1} \beta_j^{(im)}\,|\hat{v}_j^{(m)}\rangle\,, \tag{12}$$

then we perform the modified QSVE $U_{\text{SVE}}^{(m)}$ on $\hat{T}^{(m)}$, The detail of this operation is further illustrated in the following theorem.

Theorem 4. *Given every frontal slice of $\mathcal{T}$ stored in the data structure as mentioned in Lemma 2, there is a quantum algorithm, denoted as $U_{\text{SVE}}^{(m)}$, that uses the input $|\hat{\mathcal{T}}(i,:,m)\rangle$ in the form (12) and outputs the state*

$$\sum_j \beta_j^{(im)}\,|\hat{v}_j^{(m)}\rangle^d\,|\overline{\hat{\sigma}}_j^{(m)}\rangle^b \tag{13}$$

with probability at least $1 - 1/\text{poly}(N)$, where $\hat{v}_j^{(m)}$ is the right singular vector of $\hat{T}^{(m)}$, and $\overline{\hat{\sigma}}_j^{(m)}$ is an estimate of $\hat{\sigma}_j^{(m)}$ satisfying $|\overline{\hat{\sigma}}_j^{(m)} - \hat{\sigma}_j^{(m)}| \leq \epsilon_{\text{SVE}}^{(m)}||\hat{T}^{(m)}||_F$. The running time to implement $U_{\text{SVE}}^{(m)}$ is $\mathcal{O}\left(N\text{polylog}N/\epsilon_{\text{SVE}}^{(m)}\right)$.

Proof. *See the proof in Appendix A.*

Remark 2. *It should be noticed that the process $U_{\text{SVE}}^{(m)}$ proposed in Theorem 4 is distinct from the traditional QSVE technique stated in Section 2.3. Specifically, Theorem 4 shows that we can estimate the singular values of $\hat{T}^{(m)}$ if every frontal slice of original subsample tensor $T^{(k)}$ is stored in the data structure, $m, k = 0,\cdots,N-1$. The proof of Theorem 4 presents a detailed illustration of the procedure of $U_{\text{SVE}}^{(m)}$.*

Similar with the quantum singular value decomposition for matrices [38] that the output allows singular values and associated singular vectors to be revealed in a quantum form, the state in (13) also finds the estimated singular values of $\hat{T}^{(m)}$ and store them in the third register, superposed with corresponding singular vectors. Therefore, combining (11) and (13), the state after **Step 2** is

$$\frac{1}{||\mathcal{T}(i,:,:)||_F} \sum_{m,j} ||\hat{\mathcal{T}}(i,:,m)||_2 \beta_j^{(im)}\,|\hat{v}_j^{(m)}\rangle^d\,|\overline{\hat{\sigma}}_j^{(m)}\rangle^b\,|m\rangle^e \triangleq |\xi_1\rangle\,, \tag{14}$$

In **Steps 3-5**, our goal is to project each tube $\hat{\mathcal{T}}(i,:,m)$ onto the subspace spanned by the right singular vectors $\hat{v}_j^{(m)}$ corresponding to singular values greater than the threshold τ_m. As shown in FIG. 3, in Step 3, we first add an ancillary register $|0\rangle^a$ and then apply a unitary operator

$$V = \sum_{m=0}^{N-1} V^{(m)} \otimes |m\rangle\,\langle m|^e \tag{15}$$

acting on the registers b and a controlled by the register e, where $V^{(m)}$ is a 2-qubit conditional rotation gate that maps $|h\rangle^b\,|0\rangle^a \to |h\rangle^b\,|1\rangle^a$ if $h < \tau_m$ and $|h\rangle^b\,|0\rangle^a \to |h\rangle^b\,|0\rangle^a$ otherwise. Therefore, after **Step 3**, we get

$$|\xi_2\rangle = \frac{1}{||\mathcal{T}(i,:,:)||_F} \sum_{m=0}^{N-1} ||\hat{\mathcal{T}}(i,:,m)||_2 \left(\sum_{j:\overline{\hat{\sigma}}_j^{(m)} \geq \tau_m} \beta_j^{(im)}\,|\hat{v}_j^{(m)}\rangle^d\,|\overline{\hat{\sigma}}_j^{(m)}\rangle^b\,|0\rangle^a + \sum_{j:\overline{\hat{\sigma}}_j^{(m)} < \tau_m} \beta_j^{(im)}\,|\hat{v}_j^{(m)}\rangle^d\,|\overline{\hat{\sigma}}_j^{(m)}\rangle^b\,|1\rangle^a \right) |m\rangle^e\,. \tag{16}$$

In **Step 4**, we apply the inverse modified QSVE which is denoted as $U_{\text{SVE}}^{(m)\dagger}$ in FIG. 3, and discard the register b. Then we measure line a, i.e. the last register of (16), and postselect the outcome $|0\rangle^a$, getting

$$|\xi_3\rangle = \frac{1}{\alpha} \sum_{m=0}^{N-1} \sum_{j,\geq\tau_m} \beta_j^{(im)}||\hat{\mathcal{T}}(i,:,m)||_2\,|\hat{v}_j^{(m)}\rangle^d\,|m\rangle^e\,, \tag{17}$$

where $\alpha = \sqrt{\sum_{m=0}^{N-1} \sum_{j,\geq \tau_m} ||\hat{\mathcal{T}}(i,:,m)||_2^2 \cdot |\beta_j^{(im)}|^2}$. The probability that we obtain the outcome $|0\rangle$ in **Step 4** is

$$\frac{||\hat{\mathcal{T}}_{\geq \boldsymbol{\tau}}(i,:,:)||_F^2}{||\mathcal{T}(i,:,:)||_F^2}, \tag{18}$$

where $\hat{\mathcal{T}}_{\geq \boldsymbol{\tau}}$ denotes the tensor whose m-th frontal slice is $\hat{T}_{\geq \tau_m}^{(m)}$ obtained by truncating $\hat{T}^{(m)}$ with threshold τ_m. Hence, based on amplitude amplification, we have to repeat the measurement $\mathcal{O}\left(\frac{||\mathcal{T}(i,:,:)||_F}{||\hat{\mathcal{T}}_{\geq \boldsymbol{\tau}}(i,:,:)||_F}\right)$ times to ensure the success probability of getting the outcome $|0\rangle$ is close to 1.

Comparing (12) with (17), we find that the unnormalized state $\sum_{j,\geq \tau_m} \beta_j^{(im)} ||\hat{\mathcal{T}}(i,:,m)||_2 \, |\hat{v}_j^{(m)}\rangle$ can be seen as an approximation of $\hat{\mathcal{T}}(i,:,m)$, $m = 0, \cdots, N-1$. Hence, $|\xi_3\rangle$ corresponds to an approximation of $|\hat{\mathcal{T}}(i,:,:)\rangle$.

In **Step 5**, we perform the inverse QFT, denoted as $F^\dagger$ in line e of FIG. 3, on $|\xi_3\rangle$ in (17) to get the final state

$$\begin{aligned}|\xi_4\rangle = & \frac{1}{\alpha\sqrt{N}} \sum_{t,m=0}^{N-1} \sum_{j,\geq \tau_m} \beta_j^{(im)} \omega^{-tm} \\ & ||\hat{\mathcal{T}}(i,:,m)||_2 \, |\hat{v}_j^{(m)}\rangle^d \, |t\rangle^e ,\end{aligned} \tag{19}$$

which corresponds to an approximation of $\mathcal{T}(i,:,:)$, and thus it can also be regarded as an approximation of $\mathcal{A}(i,:,:)$.

In **the last step**, user i is recommended a product j varying with different contexts as needed by measuring the output state $|\xi_4\rangle$. For example, if we need the recommended index at a certain context t_0, we can first measure the last register of $|\xi_4\rangle$ in the computational basis and postselect the outcome $|t_0\rangle$ in line e, as is shown in FIG. 3, obtaining the state propositional to (unnormalized)

$$\sum_{m=0}^{N-1} \sum_{j,\geq \tau_m} \beta_j^{(im)} \omega^{-t_0 m} ||\hat{\mathcal{T}}(i,:,m)||_2 \, |\hat{v}_j^{(m)}\rangle^d . \tag{20}$$

We next measure this state in the computational basis to get an index j which is proved to be a good recommendation for user i at context t_0.

Algorithm 2 is summarized below, whose circuit is shown in FIG. 3.

Algorithm 2 Quantum algorithm for recommendation systems modeled by third-order tensors

```
Require: Assumption 1, a user index $i$, the state $|\mathcal{T}(i,:,:)\rangle$ corresponding to the preference information of user $i$, precision $\epsilon_{\text{SVE}}^{(m)}$, the truncation threshold $\tau_m$, $m = 0, \cdots, N-1$, and a context $t_0$.
Output: the recommended index $j$ for the user $i$ at the context $t_0$.
1: Perform the QFT on the last register of the input state $|\mathcal{T}(i,:,:)\rangle$, to obtain $|\hat{\mathcal{T}}(i,:,:)\rangle$ in (8).
2: Perform the modified QSVE on the matrix $\hat{T}^{(m)}$ parallelly, using the input $|\hat{\mathcal{T}}(i,:,:)\rangle$ with precision $\epsilon_{\text{SVE}}^{(m)}$, $m = 0, \cdots, N-1$, to get the state $|\xi_1\rangle$ defined in (14).
3: Add an ancilla qubit $|0\rangle^a$ and apply a unitary transformation $V$ in (15) to obtain the state $|\xi_2\rangle$ in (16).
4: Apply the inverse modified QSVE and discard the register $b$, then measure the ancilla register $a$ in the computational basis and postselect the outcome $|0\rangle$, then delete the register $a$, obtaining the state $|\xi_3\rangle$ in (17).
5: Perform the inverse QFT on the register $e$, to get $|\xi_4\rangle$ in (19).
6: Measure the register $e$ in the computational basis and postselect the outcome $|t_0\rangle$. Then measure the register $d$ in the computational basis to get the index $j$.
```

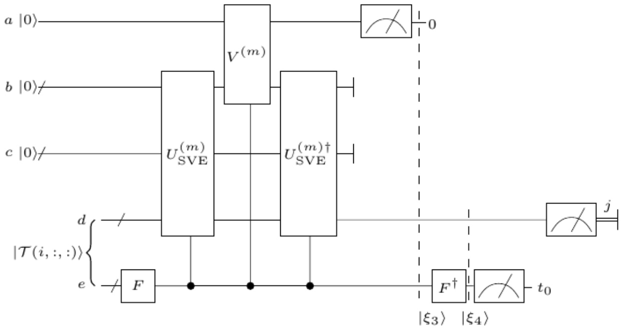

Figure 3: Circuit for Algorithm 2, and the process of $U_{\text{SVE}}^{(m)}$ is given in the proof of Theorem 4 in Appendix A.

3.3 Error analysis

In this section, the i-th horizontal slice of the tensor $\mathcal{T}_{\geq \boldsymbol{\tau}}$ is proved to be an approximation of $\mathcal{A}(i,:,:)$.

Then sampling from the matrix $\mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)$ yields good recommendations for user i; see Theorem 5. The following two lemmas, Lemmas 4 and 5, are used in the proof of Theorem 5. The proofs of Lemma 5 and Theorem 5 can be found in Appendices B and C respectively.

Lemma 4. *[18] Let $\tilde{A}$ be an approximation of the matrix A such that $||A-\tilde{A}||_F \leq \epsilon||A||_F$. Then, the probability that sampling from $\tilde{A}$ provides a bad recommendation is*

$$\Pr_{(i,j)\sim\tilde{A}}[(i,j)\mathrm{bad}] \leq \left(\frac{\epsilon}{1-\epsilon}\right)^2. \tag{21}$$

Lemma 5. *Let $A\in\mathbb{R}^{N\times N}$ be a matrix and A_k be the best rank-k approximation such that $||A-A_k||_F \leq \epsilon||A||_F$. If the threshold for truncating the singular values of A is chosen as $\sigma=\frac{\epsilon||A||_F}{\sqrt{k}}$, then*

$$||A-A_{\geq\sigma}||_F \leq 2\epsilon||A||_F. \tag{22}$$

Theorem 5. *Let Assumption 1 holds. For each $m=0,\ldots,N-1$, assume $\epsilon^{(m)}$ satisfies $||\hat{T}^{(m)}-\hat{T}_k^{(m)}||_F \leq \epsilon^{(m)}||\hat{T}^{(m)}||_F$. Define $\tau_m=\frac{\epsilon^{(m)}||\hat{T}^{(m)}||_F}{\sqrt{k}}$. Algorithm 2 outputs the state $|\mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)\rangle$ corresponding to an approximation of $\mathcal{A}(i,:,:)$ such that there are at least $(1-\delta)N$ users, of which each user i satisfies*

$$||\mathcal{A}(i,:,:)-\mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)||_F \leq \epsilon||\mathcal{A}(i,:,:)||_F \tag{23}$$

with probability at least $p_1=1-e^{-\zeta^2\left(\frac{1}{p}-1\right)\frac{||\mathcal{A}||_F^2}{3N(1+\gamma)}}$, where $\delta,\gamma,\zeta\in(0,1)$. The precision ϵ in (23) is $\epsilon=\frac{10(1+\zeta)(1-p)\delta+11\epsilon_0(1+\gamma)}{10\delta p}$, where p is the subsampling probability and $\epsilon_0=2\max_m\epsilon^{(m)}$. Moreover, let t be chosen uniformly from 0 to $N-1$. The probability that sampling according to $\mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,t)$ (equivalent to measuring the state $|\mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,t)\rangle$ in the computational basis) provides a bad recommendation is

$$\Pr_{t\sim\mathcal{U}_N,j\sim\mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,t)}[(i,j,t)\mathrm{bad}] \leq \left(\frac{\epsilon}{1-\epsilon}\right)^2 \tag{24}$$

It should be noted that by taking reasonable values for the parameters δ,γ,ζ,p, our algorithm can produce good recommendations with high probability, in other words, the probability p_1 is close to 1 and the precision ϵ could be comparatively small.

3.4 Complexity analysis

The complexity of Algorithm 2 is given by the following result.

Theorem 6. *With notation given in Theorem 5, for at least $(1-\delta)N$ users, Algorithm 2 outputs an approximation state of $|\mathcal{A}(i,:,:)\rangle$ with time complexity $\mathcal{O}\left(\frac{\sqrt{k}N\mathrm{polylog}(N)(1+\gamma)}{\min_m\epsilon^{(m)}(1+\epsilon)\sqrt{p}}\right)$. For suitably chosen values of parameters δ,ζ,γ,p and $\epsilon^{(m)}$, the running time of Algorithm 2 is $\mathcal{O}\left(\sqrt{k}N\mathrm{polylog}(N)\right)$.*

The proof of Theorem 6 can be found in Appendix D.

4 Simulations

In this section, we numerically validate our model with recommendation systems tasks. As we do not have a large-scale quantum computer, to validate our quantum algorithm, we investigate the performance of its classical counterpart, namely, truncated t-svd [19], on real datasets in the classical computer. Due to the closeness, the testing results should be true for a fault-tolerant quantum computer.

4.1 Experimental setting

We choose three multiverse recommendation systems algorithms that are based on tensor decompositions: T-HOSVD [21], TT decomposition [29], a Collaborative Filtering method (TF & SGD) [16], and compare them to truncated t-svd. All the experiments are performed under Windows 10, Python 3.7 and MATLAB R2016b running on a desktop (Intel Core i7 @ 3.60 GHz, 32.0G RAM). In each experiment we repeat 10 times and average the results.

In order to evaluate the performance of different methods, we compare the relative square error (RSE) defined in dB, mean absolute error (MAE), and root mean square error (RMSE), which are defined as follows

$$\mathrm{RSE}=20\log_{10}(\|\mathcal{T}_{\geq\tau}-\mathcal{A}\|_F/\|\mathcal{A}\|_F), \tag{25}$$

$$\mathrm{MAE}=\frac{1}{K}\sum_{i,j,k}\mathcal{D}_{ijk}\left|(\mathcal{T}_{\geq\tau})_{ijk}-\mathcal{A}_{ijk}\right|, \tag{26}$$

$$\mathrm{RMSE}=\|\mathcal{T}_{\geq\tau}-\mathcal{A}\|_F/\sqrt{K}, \tag{27}$$

where K is the total number of observed ratings, $\mathcal{D} \in \{0,1\}^{N_1 \times N_2 \times N_3}$ is a binary tensor with nonzero entries D_{ijk} whenever $\mathcal{A}_{ijk}$ is observed. These three measures are widely used in the recommendation systems literature.

4.2 Data

We test these four algorithms on two real datasets: Yahoo! Webscope Movies and Movielens, which are described as follows:

- For Yahoo! Webscope Movies dataset[1] with 7642 users, 11915 movies and 221K ratings in a $\{1, \cdots, 5\}$ scale, we select the first 800 users and 4623 corresponding movies. Besides, the original Yahoo! Webscope Movies dataset contains user age and gender features. We choose user's year of birth as the third dimension and consider it as the context variable. Therefore, the size of resulting tensor is $800 \times 4623 \times 51$ with 23782 nonzero entries.

- Movielens[2] is the benchmark dataset in recommendation systems. Here we choose the "Movielens-latest-small" dataset, which has 610 users, 9742 movies and 100K ratings with a timestamp. For preprocessing the data, we divide the timestamps that users give ratings into 60 timeslots, obtaining a third-order tensor $\mathcal{A} \in \mathbb{R}^{610 \times 9742 \times 60}$ in which the three modes represents users, movies, and time respectively.

4.3 Comparison result

In FIGs. 4 and 5, the RSE, MAE, and RMSE values obtained by different methods are plotted for Yahoo! Webscope Movies and Movielens datasets respectively with 80% sampling probability. We first compare truncated t-svd, T-HOSVD, and TT methods since they do not require optimization and they all follow similar procedures. T-HOSVD is composed of a core tensor and unitary matrices storing the principal compoments of each mode. TT decomposition represents a tensor as the link structure of each core tensor. The truncation rank is (k, k, k) for k ranging from 5 to 50. Here, (k, k, k) refers to the multi-rank of t-svd, the multilinear rank of T-HOSVD, and the TT rank of TT decomposition. We can observe that the RSE values obtained by truncated t-svd are much lower than those obtained by T-HOSVD and TT decompositions. When k is greater than 20 (for Yahoo!) or 8 (for Movielens), the RMSE and MAE values of truncated t-svd are also lower than T-HOSVD and TT.

Moreover, to validate the effectiveness of our algorithm, we also evaluate the average probability of providing a bad recommendation, denoted as Pr[bad], when we sample according to $\mathcal{T}_{\geq \tau}$. Here, we calculate Pr[bad] by

$$\Pr[\text{bad}] = \frac{1}{N} \sum_{i=0}^{N-1} \left(\frac{\epsilon_i}{1-\epsilon_i} \right)^2, \tag{28}$$

where $\epsilon_i = \|\mathcal{T}_{\geq \tau}(i,:,:) - \mathcal{A}(i,:,:)\|_F / \|\mathcal{A}(i,:,:)\|_F$ and N refers to the number of users in the dataset. Our theoretical basis is Lemma 4 and Theorem 5. The curves of Pr[bad] are plotted in the bottom right of FIGs. 4 and 5. In summary, truncated t-svd has better performance than T-HOSVD and TT when applied to context-aware recommendation systems datasets.

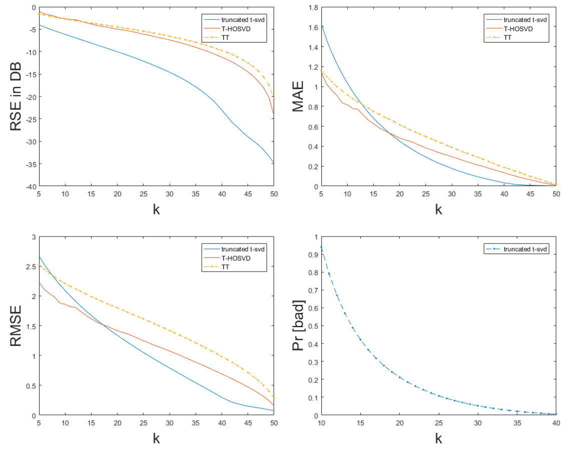

Figure 4: The comparison results on Yahoo! Webscope Movies dataset. The top-left, top-right and bottom-left figures plot the RSE, MAE, RMSE against truncation rank k respectively. The bottom-right figure shows that the average probability of providing a bad recommendation.

[1] Webscope v1.0, http://research.yahoo.com/

[2] grouplens.org/datasets/movielens

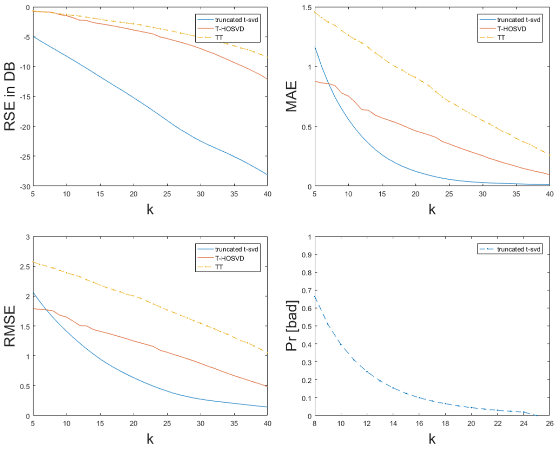

Figure 5: The comparison results on Movielens dataset.

In [16], the authors introduce a Collaborative Filtering method based on Tensor Factorization (TF & SGD) in Multiverse Recommendation model. To train this model, they minimize the regularized risk function by stochastic gradient descent (SGD). Due to the use of SGD algorithm, the computational cost is increased and the algorithm can be sensitive to initial guess. We compare the MAE values and average running time of truncated t-svd and TF & SGD on Yahoo! Webscope Movies dataset; see FIG. 6. For TF & SGD, the regularization parameters for factor matrices and core tensor are all set as 0.1, and their initial learning rates are all set as 0.001. The multilinear rank is set as $d_U = d_M = d_C = k$. We can observe that the MAE values of TF & SGD is comparatively stable with k. When $k \geq 14$, the truncated t-svd performs better than TF & SGD in MAE. Moreover, the average running time of truncated t-svd is only around one fifth to one third of the time of TF & SGD.

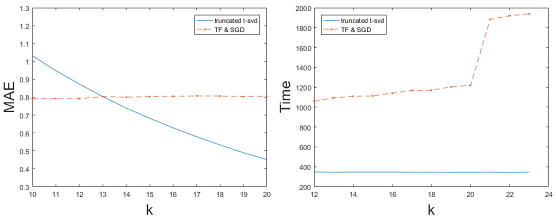

Figure 6: The MAE and average running time of truncated t-svd and TF & SGD algorithms on Yahoo! Webscope Movies dataset.

5 Relations with Tang's algorithms

In this section, we compare our 3D quantum algorithm with Tang's 2D quantum-inspired recommendation systems algorithm. For a 2D recommendation system modeled by an $N \times N$ matrix, Tang's quantum-inspired algorithm [47] was shown to have complexity $\mathcal{O}(\text{poly}(k)\text{polylog}(N))$, an exponential speedup compared to other classical methods, only polynomially slower than Kerenidis and Prakash's quantum algorithm [18]. This raised the question of whether Tang's algorithm was actually useful in practice. In order to study its practical performance, Lloyd et al. investigated Tang's 2D recommendation systems algorithm [2]. They commented that Tang's algorithm is only advantageous for preference matrix with extremely large dimension, very low rank and low condition number. However, it remains unclear whether such kind of datasets actually exists in practice. They also found that Tang's algorithm takes more time and suffers higher inaccuracies than the classical exact diagonalization method. Kerenidis and Prakash [17] also commented that this high dependence on the rank and other parameters makes Tang's algorithm impractical. Therefore, it is unlikely that Tang's algorithm can replace Kerenidis and Prakash's algorithm in practical 2D recommendation systems.

Moreover, both Tang and Kerenidis and Prakash's algorithms are for 2D recommendation systems, and our proposed quantum algorithm, Algorithm 2, is for 3D recommendation systems. To our best knowledge, there is no previous work on quantum-inspired classical algorithm of 3D recommendation systems. Clearly, this is a very interesting and practically important problem. Hence, here we would like to share our understandings on how to extend Tang's quantum-inspired techniques to third-order recommendation systems.

To our understanding, there are at least two difficulties in extending quantum-inspired classical algorithms from 2D to 3D recommendation systems. First, as the product that a user prefers in a certain context is very likely to affect the recommendation for him/her at other contexts, the relations among different frontal slices (a matrix of users × products) of the preference tensor should be taken into account. Our quantum algorithm addresses this problem very well because the QFT is performed to combine a user's preferences in different contexts. On the other hand, if we extend Tang' idea to third-order recommendation systems, the immediate idea is to apply Tang's 2D recommendation systems algorithm to every frontal slice of the preference tensor, but this idea fails to consider the preference correlations among different contexts for a certain user. In order to take this factor into consideration, the rough idea is to apply DFT to each horizontal slice of the preference tensor in order to achieve the same effect as QFT. It is easy to see that the complexity of this step is $\mathcal{O}(N^3\log(N))$. If we choose any unitary transformation to substitute DFT, which is adopted in the transformed t-svd [45], the complexity will be $\mathcal{O}(N^{3.3})$. In either case, the complexity of this step alone exceeds that of our quantum algorithm for 3D recommendation systems. Therefore, how to control the complexity of quantum-inspired classical algorithms for 3D recommendation systems is the first obstacle we need to overcome. Another difficulty is how to guarantee that the designed quantum-inspired algorithm is useful in practice, or suitable for real datasets.

6 Conclusion

In this paper, we proposed the first quantum algorithm for context-aware recommendation systems modeled by third-order tensors. Moreover, we showed that this quantum algorithm can provide good recommendations varying with contexts and run in expected time $\mathcal{O}(\sqrt{k}N\text{polylog}(N))$ for some suitable parameters. The numerical simulation validates our algorithm.

Acknowledgement. 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.

A The proof of Theorem 4

Before proving Theorem 4, we would like to sketch the proof first. According to Assumption 1, each frontal slice $T^{(k)}$ is stored in the binary tree structure. Hence, based on the proof of Lemma 3 in [18], the states $|\mathcal{T}(i,:,k)\rangle$, corresponding to the i-th row of $T^{(k)}$, can be prepared efficiently by operators P_k. Based on these operators, two new isometries $\hat{P}_m$ and $\hat{Q}_m$ are constructed in order to perform QSVE on $\hat{T}^{(m)}$. The detail of the proof is given below.

Proof. Since every $T^{(k)}$, $k = 0, \cdots, N-1$, is stored in the binary tree structure, the quantum computer can perform the following mapping in $\mathcal{O}(\text{polylog}(N))$ time, as shown in Theorem 5.1 in [18]:

$$\begin{aligned} U_{P_k} &: |i\rangle\,|0\rangle \\ &\to |i\rangle\,|\mathcal{T}(i,:,k)\rangle = \frac{1}{||\mathcal{T}(i,:,k)||_2}\sum_{j=0}^{N-1}\mathcal{T}_{ijk}\,|i\rangle\,|j\rangle\,, \end{aligned} \tag{29}$$

where $\mathcal{T}(i,:,k)$ is the i-th row of $T^{(k)}$.

Define the degenerate operator $P_k \in \mathbb{R}^{N^2\times N}$ related to U_{P_k} as

$$P_k : \quad |i\rangle \to |i\rangle\,|\mathcal{T}(i,:,k)\rangle\,. \tag{30}$$

That is,

$$P_k = \sum_{i=0}^{N-1}|i\rangle\,|\mathcal{T}(i,:,k)\rangle\,\langle i|\,. \tag{31}$$

Based on the efficiently implemented operator U_{P_k}, we define another operator

$$U_{\hat{P}_m} \triangleq \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\omega^{km}U_{P_k}. \tag{32}$$

It can be easily seen that the operator $U_{\hat{P}_m}$ achieves the state preparation of the rows of the matrix $\hat{T}^{(m)}$, i.e., $U_{\hat{P}_m} = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\sum_{i=0}^{N-1}\omega^{km}\,|i\rangle\,|\mathcal{T}(i,:,k)\rangle\,\langle i|\,\langle 0| = \sum_i |i\rangle\,|\hat{\mathcal{T}}(i,:,m)\rangle\,\langle i|\,\langle 0|$, where $|\hat{\mathcal{T}}(i,:,m)\rangle$ is the state of the i-th row of $\hat{T}^{(m)}$. Similarly, the isometry corresponding to $U_{\hat{P}_m}$ is $\hat{P}_m = \sum_i |i\rangle\,|\hat{\mathcal{T}}(i,:,m)\rangle\,\langle i|$. It can be easily shown that $\hat{P}_m$ is an isometry since $\hat{P}_m^\dagger\hat{P}_m = I_N$. Since U_{P_k} can be implemented in time $\mathcal{O}(\text{polylog}(N))$, $U_{\hat{P}_m}$ can be implemented in time $\mathcal{O}(N\text{polylog}N)$ using the linear combination of unitaries (LCU) technique [6, 43, 24, 44, 23].

The LCU technique was first proposed by Long in their work [23] in a more general form, and Shao summarized this result in [44]. The problem of

LCU can be formulated as follows: Given $\alpha_i \in \mathbb{C}$ and unitary operators U_i, $i = 0, 1, \cdots, N-1$, implement linear operator $L = \sum_{j=0}^{N-1} \alpha_j U_j$. The algorithm stated in [6] implements L in time $\mathcal{O}((T_{\text{in}} + \log N) N \max_j |\alpha_j| / ||L|\psi\rangle||)$, where $|\psi\rangle$ is any given initial state and T_{in} is the time to implement $U_0, U_1, \cdots, U_{N-1}$. In our case, $T_{\text{in}} = N\text{polylog}N$, $\alpha_j = \omega^{km}/\sqrt{N}$ and the input state is chosen as $|\psi\rangle = \sum_{i=0}^{N} |i\rangle|0\rangle$. Thus, $||L|\psi\rangle|| = ||\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\omega^{km}U_{P_k}|\psi\rangle|| = ||\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\sum_{i=0}^{N-1}\omega^{km}|i\rangle|\mathcal{T}(i,:,k)\rangle|| = \sqrt{N}$. Therefore, the complexity to implement $U_{\hat{P}_m}$ is $\mathcal{O}(N\text{polylog}N)$.

Next, we define the mapping

$$\begin{aligned} U_{\hat{Q}_m} : |0\rangle|j\rangle \to & |\boldsymbol{s}_{\hat{T}^{(m)}}\rangle|j\rangle \\ = & \frac{1}{||\hat{T}^{(m)}||_F}\sum_i ||\hat{\mathcal{T}}(i,:,m)||_2 |i\rangle|j\rangle, \end{aligned} \tag{33}$$

where $\boldsymbol{s}_{\hat{T}^{(m)}}$ is a vector whose i-th entry is $\frac{||\hat{\mathcal{T}}(i,:,m)||}{||\hat{T}^{(m)}||_F}$. Similar with $U_{\hat{P}_m}$, the corresponding isometry is defined as $\hat{Q}_m = \sum_j |\boldsymbol{s}_{\hat{T}^{(m)}}\rangle|j\rangle\langle j|$ satisfying $\hat{Q}_m^\dagger\hat{Q}_m = I_N$, which can be verified easily.

Now we perform QSVE on the matrix $\hat{T}^{(m)}$. First, the factorization $\frac{\hat{T}^{(m)}}{||\hat{T}^{(m)}||_F} = \hat{P}_m^\dagger\hat{Q}_m$ can be easily verified. Second, we can prove that $2\hat{P}_m\hat{P}_m^\dagger - I_{N^2}$ is a reflection and it be implemented through $U_{\hat{P}_m}$. Actually,

$$\begin{aligned} & 2\hat{P}_m\hat{P}_m^\dagger - I_{N^2} \\ = & 2\sum_i |i\rangle|\hat{\mathcal{T}}(i,:,m)\rangle\langle i|\langle\hat{\mathcal{T}}(i,:,m)| - I_{N^2} \\ = & U_{\hat{P}_m}\left[2\sum_i |i\rangle|0\rangle\langle i|\langle 0| - I_{N^2}\right]U_{\hat{P}_m}^\dagger, \end{aligned} \tag{34}$$

where $2\sum_i |i\rangle|0\rangle\langle i|\langle 0| - I_{N^2}$ is a reflection. The similar result holds for $2\hat{Q}_m\hat{Q}_m^\dagger - I_{N^2}$.

Now denote

$$W_m = \left(2\hat{P}_m\hat{P}_m^\dagger - I_{N^2}\right)\left(2\hat{Q}_m\hat{Q}_m^\dagger - I_{N^2}\right). \tag{35}$$

Let $\hat{T}^{(m)} = \sum_{i=0}^{r-1}\hat{\sigma}_i^{(m)}\hat{u}_i^{(m)}\hat{v}_i^{(m)\dagger}$ be the singular value decomposition of $\hat{T}^{(m)}$. We can prove that the subspace spanned by $\{\hat{Q}_m|\hat{v}_i^{(m)}\rangle, \hat{P}_m|\hat{u}_i^{(m)}\rangle\}$ is invariant under the unitary transformation W_m:

$$W_m\hat{Q}_m|\hat{v}_i^{(m)}\rangle = \frac{2\hat{\sigma}_i^{(m)}}{||\hat{T}^{(m)}||_F}\hat{P}_m|\hat{u}_i^{(m)}\rangle - Q|\hat{v}_i^{(m)}\rangle,$$

$$\begin{aligned} & W_m\hat{P}_m|\hat{u}_i^{(m)}\rangle = \\ & \left(\frac{4\hat{\sigma}_i^2}{||\hat{T}^{(m)}||_F} - 1\right)\hat{P}_m|\hat{u}_i^{(m)}\rangle - \frac{2\hat{\sigma}_i}{||\hat{T}^{(m)}||_F}\hat{Q}_m|\hat{v}_i^{(m)}\rangle. \end{aligned}$$

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 $|\omega_{i\pm}^{(m)}\rangle$ with correspondent eigenvalues $e^{\pm i\theta_i^{(m)}}$, where $\theta_i^{(m)}$ is the rotation angle satisfying

$$\cos(\theta_i^{(m)}/2) = \frac{\hat{\sigma}_i^{(m)}}{||\hat{T}^{(m)}||_F}, \tag{36}$$

that is,

$$\begin{aligned} \hat{Q}_m|\hat{v}_i^{(m)}\rangle &= \sqrt{2}\left(|\omega_{i+}^{(m)}\rangle + |\omega_{i-}^{(m)}\rangle\right), \\ \hat{P}_m|\hat{u}_i^{(m)}\rangle &= \sqrt{2}\left(e^{i\theta_i/2}|\omega_{i+}^{(m)}\rangle + e^{-i\theta_i/2}|\omega_{i-}^{(m)}\rangle\right). \end{aligned}$$

Here, we choose the input state $|\hat{\mathcal{T}}(i,:,m)\rangle$ represented in (12), then

$$\hat{Q}_m|\hat{\mathcal{T}}(i,:,m)\rangle = \sum_{j=0}^{N-1}\sqrt{2}\beta_j^{(im)}\left(|\omega_{i+}^{(m)}\rangle + |\omega_{i-}^{(m)}\rangle\right). \tag{37}$$

Performing the phase estimation on W_m with running time $\mathcal{O}\left(N\text{polylog}N/\epsilon_{\text{SVE}}^{(m)}\right)$, and computing the estimated singular value of $\hat{T}^{(m)}$ through oracle with a computable function $f(x) = ||\hat{T}^{(m)}||_F\cos(x/2)$. According to the relations between $\theta_i^{(m)}$ and $\hat{\sigma}_i^{(m)}$ in (36), we obtain

$$\sum_{j=0}^{N-1}\sqrt{2}\beta_j^{(im)}\left(|\omega_{i+}^{(m)}\rangle|\overline{\theta}_i^{(m)}\rangle + |\omega_{i-}^{(m)}\rangle|-\overline{\theta}_i^{(m)}\rangle\right)|\overline{\sigma}_i^{(m)}\rangle. \tag{38}$$

we next uncompute the phase estimation procedure and then apply the inverse of $U_{\hat{Q}_m}$ to obtain the desired state (13) in Theorem 4. □

B The proof of Lemma 5

Proof. Let σ_i denote the singular value of A and l be the largest integer for which $\sigma_l \geq \frac{\epsilon||A||_F}{\sqrt{k}}$. By the triangle inequality, $||A - A_{\geq\sigma}||_F \leq ||A - A_k||_F + ||A_k - A_{\geq\sigma}||_F$. If $k \leq l$, it's easy to conclude that

$||A_k - A_{\geq\sigma}||_F \leq ||A - A_k||_F \leq \epsilon||A||_F$. If $k > l$, $||A_k - A_{\geq\sigma}||_F^2 = \sum_{i=l+1}^k \sigma_i^2 \leq k\sigma_{l+1}^2 \leq k\left(\frac{\epsilon||A||_F}{\sqrt{k}}\right)^2 \leq (\epsilon||A||_F)^2$. In either case, we have $||A - A_{\geq\sigma}||_F \leq 2\epsilon||A||_F$. □

C The proof of Theorem 5

Proof. Based on Lemma 5 in the main text, if the best rank-k approximation satisfies $||\hat{T}^{(m)} - \hat{T}_k^{(m)}||_F \leq \epsilon^{(m)}||\hat{T}^{(m)}||_F$, then

$$||\hat{T}^{(m)} - \hat{T}_{\geq\tau_m}^{(m)}||_F \leq 2\epsilon^{(m)}||\hat{T}^{(m)}||_F \leq \epsilon_0||\hat{T}^{(m)}||_F \tag{39}$$

for $m = 0, \cdots, N-1$. By summing both sides of (39) over m, we get

$$||\hat{\mathcal{T}} - \hat{\mathcal{T}}_{\geq\boldsymbol{\tau}}||_F^2 = \sum_{m=0}^{N-1} ||\hat{T}^{(m)} - \hat{T}_{\geq\tau_m}^{(m)}||_F^2 \leq \epsilon_0^2||\hat{\mathcal{T}}||_F^2. \tag{40}$$

Since the inverse QFT along the third mode of the tensor $\mathcal{T}$ cannot change the Frobenius norm of its horizontal slice, (40) can be be re-written as

$$||\mathcal{T} - \mathcal{T}_{\geq\boldsymbol{\tau}}||_F^2 \leq \epsilon_0^2||\mathcal{T}||_F^2. \tag{41}$$

Moreover, noticing that $||\mathcal{T} - \mathcal{T}_{\geq\boldsymbol{\tau}}||_F^2 = \sum_{i=0}^{N-1} ||\mathcal{T}(i,:,:) - \mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)||_F^2$, we have $\mathrm{E}\left(||\mathcal{T}(i,:,:) - \mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)||_F^2\right) \leq \frac{\epsilon_0^2||\mathcal{T}||_F^2}{N}$. Due to Markov's Inequality ([41, Proposition 2.6]), for $\delta \in (0,1)$,

$$\begin{aligned}&\Pr\left(||\mathcal{T}(i,:,:) - \mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)||_F^2 > \frac{\epsilon_0^2||\mathcal{T}||_F^2}{\delta N}\right)\\ \leq&\frac{\mathrm{E}\left(||\mathcal{T}(i,:,:) - \mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)||_F^2\right)\delta N}{\epsilon_0^2||\mathcal{T}||_F^2} \leq \delta\end{aligned} \tag{42}$$

holds. That means at least $(1-\delta)N$ users i satisfy

$$||\mathcal{T}(i,:,:) - \mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)||_F^2 \leq \frac{\epsilon_0^2||\mathcal{T}||_F^2}{\delta N}. \tag{43}$$

Notice $\mathrm{E}\left(||\mathcal{T}||_F^2\right) = ||\mathcal{A}||_F^2/p$. Using the Chernoff bound, we have $\Pr\left(||\mathcal{T}||_F^2 > (1+\theta)||\mathcal{A}||_F^2/p\right) \leq e^{-\theta^2||\mathcal{A}||_F^2/3p}$ for $\theta \in [0,1]$, which is exponentially small. Here, we choose $\theta = 1/10$, then $||\mathcal{T}||_F^2 \leq 11||\mathcal{A}||_F^2/10p$.

Based on the second assumption in Assumption 1, we sum both sides of (6) for m and i respectively, obtaining

$$\frac{1}{1+\gamma}\frac{||\mathcal{A}||_F^2}{N} \leq ||\mathcal{A}(i,:,:)||_F^2 \leq (1+\gamma)\frac{||\mathcal{A}||_F^2}{N}, \tag{44}$$

and

$$\frac{1}{1+\gamma}\frac{||\mathcal{A}||_F^2}{N} \leq ||A^{(m)}||_F^2 \leq (1+\gamma)\frac{||\mathcal{A}||_F^2}{N}. \tag{45}$$

Then, (43) becomes

$$||\mathcal{T}(i,:,:) - \mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)||_F^2 \leq \frac{11\epsilon_0^2(1+\gamma)}{10\delta p}||\mathcal{A}(i,:,:)||_F^2. \tag{46}$$

Meanwhile, since

$$\mathrm{E}\left(||\mathcal{A}(i,:,:) - \mathcal{T}(i,:,:)||_F^2\right) = \left(\frac{1}{p} - 1\right)||\mathcal{A}(i,:,:)||_F^2,$$

then

$$\begin{aligned}&\Pr\left(||\mathcal{A}(i,:,:) - \mathcal{T}(i,:,:)||_F^2 > \nu||\mathcal{A}(i,:,:)||_F^2\right)\\ &\leq e^{-\zeta^2\left(\frac{1}{p}-1\right)\frac{||\mathcal{A}||_F^2}{3N(1+\gamma)}},\end{aligned} \tag{47}$$

where $\nu = (1+\zeta)\left(\frac{1}{p}-1\right)$ and $\zeta \in (0,1)$. That means with probability at least p_1,

$$||\mathcal{A}(i,:,:) - \mathcal{T}(i,:,:)||_F^2 \leq \nu||\mathcal{A}(i,:,:)||_F^2. \tag{48}$$

Combining (46) and (48) together and by triangle inequality, we obtain

$$||\mathcal{A}(i,:,:) - \mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)||_F \leq \epsilon||\mathcal{A}(i,:,:)||_F. \tag{49}$$

According to Lemma 4, the probability that sampling according to $\mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)$ provides a bad recommendation is

$$\Pr_{t\sim\mathcal{U}_N, j\sim\mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,t)}[(i,j,t)\text{bad}] \leq \left(\frac{\epsilon}{1-\epsilon}\right)^2. \tag{50}$$

□

D The proof of Theorem 6

Proof. According to Theorem 4, the modified QSVE algorithm performed on the frontal slice $\hat{T}^{(m)}$ takes time $\mathcal{O}\left(N\text{polylog}(N)/\epsilon_{\text{SVE}}^{(m)}\right)$.

In Step 5 of Algorithm 2, we need to repeat the measurement $\mathcal{O}\left(\frac{||\mathcal{T}(i,:,:)||_F}{||\hat{\mathcal{T}}_{\geq\boldsymbol{\tau}}(i,:,:)||_F}\right)$ times in order to ensure the probability of getting the outcome $|0\rangle$ in this step is close to 1. For most users, we can prove that $\frac{||\mathcal{T}(i,:,:)||_F}{||\hat{\mathcal{T}}_{\geq\boldsymbol{\tau}}(i,:,:)||_F}$ is bounded and the upper bound is a constant for appropriate parameters. The proof is in the following.

Since $\mathrm{E}\left(||\mathcal{T}(i,:,:)||_F^2\right) = \frac{||\mathcal{T}(i,:,:)||_F^2}{p} \leq (1+\gamma)\frac{||\mathcal{A}||_F^2}{pN}$, then by Chernoff bound,

$$||\mathcal{T}(i,:,:)||_F^2 \leq \frac{2(1+\gamma)||\mathcal{A}||_F^2}{pN} \tag{51}$$

holds with probability close to 1. Moreover, by Theorem 5, there are at least $(1-\delta)N$ users satisfying $||\mathcal{A}(i,:,:) - \mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)||_F \leq \epsilon||\mathcal{A}(i,:,:)||_F$, then $(1+\epsilon)||\mathcal{A}(i,:,:)||_F \leq ||\mathcal{T}_{\geq\boldsymbol{\tau}}(i,:,:)||_F \leq (1+\epsilon)||\mathcal{A}(i,:,:)||_F$. Since the Frobenius norm is unchanged under the Fourier transform, we get

$$(1+\epsilon)||\hat{T}_{(i)}||_F \leq ||\hat{\mathcal{T}}_{\geq\boldsymbol{\tau}}(i,:,:)||_F \leq (1+\epsilon)||\hat{T}_{(i)}||_F. \tag{52}$$

Therefore,

$$||\hat{\mathcal{T}}_{\geq\boldsymbol{\tau}}(i,:,:)||_F^2 \geq (1+\epsilon)^2||\hat{T}_{(i)}||_F^2 \geq \frac{(1+\epsilon)^2}{1+\gamma}\frac{||\mathcal{A}||_F^2}{N}. \tag{53}$$

Combining (51) and (53) together, we can conclude that for at least $(1-\delta)N$ users, $\frac{||\mathcal{T}(i,:,:)||_F}{||\hat{\mathcal{T}}_{\geq\boldsymbol{\tau}}(i,:,:)||_F}$ is bounded, that is,

$$\frac{||\mathcal{T}(i,:,:)||_F}{||\hat{\mathcal{T}}_{\geq\boldsymbol{\tau}}(i,:,:)||_F} \leq \left(\frac{(1+\gamma)\frac{2||\mathcal{A}||_F^2}{pN}}{\frac{(1+\epsilon)^2}{1+\gamma}\frac{||\mathcal{A}||_F^2}{N}}\right)^{1/2} = \frac{\sqrt{2}(1+\gamma)}{(1+\epsilon)\sqrt{p}}. \tag{54}$$

The precision for the singular value estimation algorithm on the matrix $||\hat{T}^{(m)}||_F$ can be chosen as $\epsilon_{\text{SVE}}^{(m)} = \frac{\tau_m}{||\hat{T}^{(m)}||_F}$. Therefore, the total time complexity of Algorithm 2 is

$$(\log N)^4 \cdot \frac{N\text{polylog}(N)}{\min\limits_m \epsilon_{\text{SVE}}^{(m)}} \cdot \frac{||\mathcal{T}(i,:,:)||_F}{||\hat{\mathcal{T}}_{\geq\boldsymbol{\tau}}(i,:,:)||_F}$$

$$\leq (\log N)^4 N\text{polylog}(N) \max_m \frac{||\hat{T}^{(m)}||_F}{\tau_m} \cdot \frac{\sqrt{2}(1+\gamma)}{(1+\epsilon)\sqrt{p}}$$

$$\approx \frac{\sqrt{k}N\text{polylog}(N)(1+\gamma)}{\min\limits_m \epsilon^{(m)}(1+\epsilon)\sqrt{p}}.$$

Here, we sort out the relations between these parameters. Given the assumed low tubal-rank k, the precision for every frontal slice after Fourier transform $\epsilon^{(m)}$ is settled, then the precision for QSVE $\epsilon_{\text{SVE}}^{(m)}$ and the truncation threshold τ_m are determined subsequently. At last, the relative error ϵ and the complexity of Algorithm 2 relate to all these parameters. □

E The quantum cost

Defining the cost of quantum circuits is not an easy task due to the fact that each quantum computer model may have a different cost for a given quantum gate. Here, the quantum cost of a reversible gate is defined to be the number of 1×1 and 2×2 reversible gates or quantum logic gates required in its design. The quantum costs of all reversible 1×1 and 2×2 gates are taken as unity [4, 42, 49, 50]. The cost of a circuit is calculated by summing up the costs of the gates composing the circuit. First, we analysize the quantum cost of QFT and QSVE circuits, then we analyze the cost of each gate in FIG. 3.

E.1 The cost of QFT

The QFT under an orthonormal basis $|x\rangle \in \{|0\rangle, \ldots, |N-1\rangle\}$ is defined as the linear operator with the following action on the basis vectors:

$$\text{QFT}: |x\rangle \to \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\omega^{x\cdot k}|k\rangle,$$

where $\omega = e^{\frac{2\pi \mathrm{i}}{N}}$. The inverse QFT is then defined as

$$\text{QFT}^\dagger: |k\rangle \to \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}\omega^{-k\cdot x}|x\rangle.$$

The circuit of QFT, shown in FIG. 7, is composed of a total number of $\mathcal{O}\left(\lceil\log N\rceil^2\right)$ H gates, CNOT gates, and controlled phase gate R_m (see [27, Section 5.1]), where

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \qquad R_m = \begin{pmatrix}1 & 0\\ 0 & e^{\frac{2\pi \mathrm{i}}{2^m}}\end{pmatrix}$$

with $m = 2, \cdots, n$, $n = \lceil\log N\rceil$.

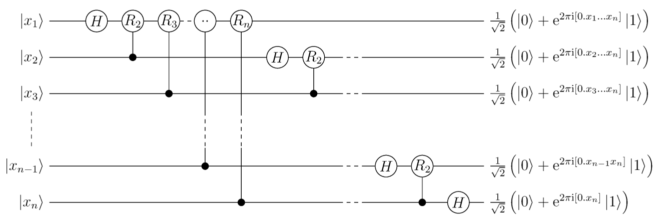
$|x_1\rangle$
$|x_2\rangle$
$|x_3\rangle$
$|x_{n-1}\rangle$
$|x_n\rangle$
H
R_2
R_3
R_n
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_1 \ldots x_n]}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_2 \ldots x_n]}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_3 \ldots x_n]}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_{n-1} x_n]}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_n]}|1\rangle\right)$

Figure 7: The circuit of QFT.

E.2 The cost of QSVE

The QSVE algorithm, originally proposed in [18], is introduced in the preliminary part of the manuscript. The QSVE algorithm and its circuit are respectively given in Algorithm 3 and FIG. 8 below. In what follows, we focus on the quantum cost of this algorithm.

Let $A \in \mathbb{R}^{N\times N}$ be a matrix with singular value decomposition $A = \sum_i \sigma_i u_i v_i^T$ stored in the binary tree data structure introduced in Lemma 2. In Step 2 of Algorithm 3, the unitary U_Q corresponding to Q is defined as

$$U_Q : |0\rangle\,|j\rangle \to |\widetilde{A}, j\rangle = \frac{1}{\|A\|_F} \sum_{i\in[N]} \|A_i\|\,|i, j\rangle \tag{55}$$

for $j = 0, \cdots, N-1$, where $\widetilde{A}$ is a vector whose i-th row is $\widetilde{A}_i = \|A_i\|$ for $i \in [N]$. Since $\widetilde{A}$ is stored in a classical binary tree with depth $\lceil \log N \rceil$, U_Q can be implemented by performing $\lceil \log N \rceil$ controlled rotations on $|0\rangle^{\otimes \lceil \log N \rceil}$ (see [18, Appendix A]). To implement each controlled rotation $\sum_{\tilde{\theta}\in\{0,1\}^{\otimes\lceil \log N \rceil}} |\tilde{\theta}\rangle\langle\tilde{\theta}| \otimes e^{-\mathrm{i}Y\tilde{\theta}}$ with Pauli Y matrix $Y = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix}$, we use one rotation controlled on each qubit of the first register which can be implemented with cost $\mathcal{O}(\lceil \log N \rceil)$ (see [52, Lemma 2]). Therefore, the cost of implementing U_Q in Step 2 is $\sum_{\lceil \log N \rceil} \lceil \log N \rceil^2$, i.e., $\mathcal{O}(\lceil \log N \rceil^3)$.

Algorithm 3 Quantum singular value estimation

Input: $A \in \mathbb{R}^{N\times N}, x \in \mathbb{R}^N$ in the data structure in Lemma 2, precision parameter $\epsilon > 0$.

```
1: Create $|x\rangle = \sum_i \alpha_i |v_i\rangle$
2: Append a first register $\left|0^{\lceil \log m \rceil}\right\rangle$ and create the
   state $|Qx\rangle = \sum_i \alpha_i |Qv_i\rangle$ as in (55).
3: Perform phase estimation with precision parameter $2\epsilon > 0$ on the input $|Qx\rangle$ for the unitary $W$
   and obtain $\sum_i \alpha_i \left|Qv_i, \overline{\theta_i}\right\rangle$.
4: Compute $\overline{\sigma_i} = \cos\left(\overline{\theta_i}/2\right)\|A\|_F$ where $\overline{\theta_i}$ is the
   estimate from phase estimation, and uncompute the output of the phase estimation.
5: Apply the inverse of the transformation in Step
   2 to obtain $\sum_i \alpha_i |v_i\rangle\,|\overline{\sigma_i}\rangle$.
```

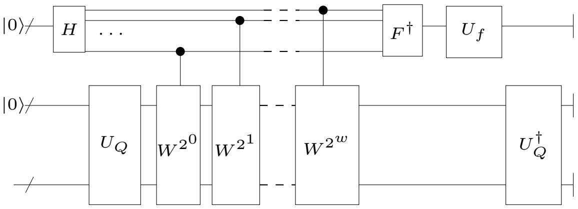

Figure 8: The circuit of QSVE algorithm. U_f is a unitary operator implemented through oracle with a computable function $f(x) = ||A||_F \cos(x/2)$. $w = \log N - 1$.

In Step 3, we can prove that the quantum cost of implementing the unitary W is $\mathcal{O}(N\lceil \log N \rceil^3)$. To be specific, notice that $W = U \cdot V$, where $V = 2QQ^T - I_{N^2}$, $U = 2PP^T - I_{N^2} = U_P\left(2\sum_i |i\rangle|0\rangle\langle i|\langle 0| - I_{N^2}\right)U_P^\dagger$, and $U_P : |i\rangle\,|0\rangle \to |i, A_i\rangle$ for $i \in [N]$. The unitary operator $2\sum_i |i\rangle|0\rangle\langle i|\langle 0| - I_{N^2}$ can be realized by the circuit shown in FIG. 9. Since each row of the matrix A is stored in the classical binary tree structure, U_P can be implemented with cost $\mathcal{O}(N\lceil \log N \rceil^3)$, as compared to the cost $\mathcal{O}(\lceil \log N \rceil^3)$ of implementing U_Q. To summarize, a total cost of $\mathcal{O}(N\lceil \log N \rceil^3)$ is required to implement W if we ignore less significant cost. Next, we perform phase estimation on W. When the singular values of W is precise to the $\lceil \log N \rceil$-th bit, the total number of rotation gate invocations is $\mathcal{O}\left(\lceil \log N \rceil^2\right)$ (See [5, Section 5.2]). The sequence of controlled-W^{2^j}, $j = 0, \cdots, \lceil \log N \rceil - 1$ operations in the phase estimation procedure can be implemented using $\mathcal{O}(\lceil \log N \rceil^3)$ gates by the modular exponentiation technique (see [27, Box 5.2]). Thus, the cost of Step 3 is $\mathcal{O}(N\lceil \log N\rceil^3)$, which dominates the cost of QSVE algorithm.

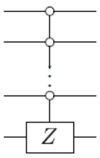

Figure 9: Quantum circuit to run $2\sum_i |i\rangle^{\otimes n}|0\rangle\langle i|^{\otimes n}\langle 0| - I^{\otimes(n+1)}$ for the input state with $n+1$ qubits. The block Z is the Pauli Z gate.

To summarize, the quantum cost of QSVE is $\mathcal{O}(N\lceil \log N\rceil^3)$.

E.3 The cost of the circuit in FIG. 3 for Algorithm 2

As discussed in Appendix E.1, the circuit of QFT in Step 1 of Algorithm 2 is composed of $\mathcal{O}(\lceil \log N\rceil^2)$ H gates, CNOT gates, and 2-qubit controlled phase gates.

In Step 2, we give a rough estimation of the cost of the controlled-unitary operator U defined in (10). Current standard methods for realizing controlled-unitary gates rely on the decomposition of $U_{\text{SVE}}^{(m)}$ into a set of 1×1 and 2×2 reversible gates. Specifically, based on the method proposed in [61], the number of additional operations required to add a control to each $U_{\text{SVE}}^{(m)}$ will generally be far less than the cost of the constructed $U_{\text{SVE}}^{(m)}$, so we next focus on estimating the cost of achieving $U_{\text{SVE}}^{(m)}, m = 0,\cdots,N-1$. The most costy step of this process lies in constructing the operator $U_{\hat{P}_m}$ in (32), $m = 0,\cdots,N-1$. According to the analysis in the paragraph above (33), implementing all U_{P_k} in (29) for $k = 0,\cdots,N-1$ occupies the major cost of achieving $U_{\hat{P}_m}, m = 0,\cdots,N-1$, which takes a total of $\mathcal{O}(N^2\lceil \log N\rceil^3)$ 1×1 and 2×2 reversible gates, since the cost of each U_{P_k} is $\mathcal{O}(N\lceil \log N\rceil^3)$ based on the cost of U_P analyzed in Appendix E.2. To summarize, the quantum cost of implementing the operator U is $\mathcal{O}(N^2\lceil \log N\rceil^3)$.

In conclusion, the quantum cost of implementing Algorithm 2 is $\mathcal{O}(N^2\lceil \log N\rceil^3)$, which is mainly concentrated in achieving the operator U if we ignore the insignificant cost, such as the cost of implementing the operator V in (29). Actually, it is reasonable that the cost of Algorithm 2 is N times more expensive than that of the QSVE algorithm proposed in [18]. The cost of the QSVE algorithm on matrix $A \in \mathbb{C}^{N\times N}$ mainly concentrates on the cost of quantum access to classical data with N rows of A stored in N binary trees. By contrast, our algorithm deals with tensor $\mathcal{T} \in \mathbb{R}^{N\times N\times N}$ whose N^2 tubes are assumed to be stored in N^2 binary trees, so our algorithm needs N times more gates than the QSVE algorithm to quantum access to the data structure.

Since implementing the physical operations depends on many factors, such as the initial quantum circuit, the quantum computer to be performed on, or the Hamiltonian of the system, the cost presented here is not necessarily the true cost but it provides a reference value. Also, it is not necessarily minimal because it can be decreased by finding more efficient quantum circuits.

The number of 1×1 and 2×2 reversible gates required by our quantum algorithm scales polynomially with the dimension of the preference tensor, which presents a big obstacle to practical realization. In fact, not only our algorithm and the QSVE algorithm, but also some benchmark algorithms, such as the HHL algorithm [13], appear too expensive to be executed efficiently by a quantum computer and are not likely to be feasible in the Noisy Intermediate-Scale Quantum (NISQ) era; see [30, Sections 6.6 and 6.7], [44, Section 1.10]. Finally, we point out most quantum machine learning algorithms focus on the time complexity rather than their quantum cost; see, e.g., [13, 36, 17, 6, 37].

References

[1] D. Achlioptas and F. McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM (JACM)*, 54(2):9, 2007.

[2] J. M. Arrazola, A. Delgado, B. R. Bardhan, and S. Lloyd. Quantum-inspired algorithms in practice. *arXiv preprint arXiv:1905.10415*, 2019.

[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[4] A. K. Biswas, M. M. Hasan, A. R. Chowdhury, and H. M. H. Babu. Efficient approaches for designing reversible binary coded decimal adders. *Microelectronics Journal*, 39(12):1693–1703, 2008.

[5] C.-F. Chiang. Quantum phase estimation with an arbitrary number of qubits. *International Journal of Quantum Information*, 11(01):1350008, 2013.

[6] B. D. Clader, B. C. Jacobs, and C. R. Sprouse. Preconditioned quantum linear system algorithm. *Physical Review Letters*, 110(25):250504, 2013.

[7] P. Comon. Tensor decompositions. *Mathematics in Signal Processing V*, pages 1–24, 2002.

[8] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[9] G. Ely, S. Aeron, N. Hao, and M. E. Kilmer. 5d seismic data completion and denoising using a novel class of tensor decompositions. *Geophysics*, 80(4):V83–V95, 2015.

[10] V. Giovannetti, S. Lloyd, and L. Maccone. Quantum random access memory. *Physical Review Letters*, 100(16):160501, 2008.

[11] L. Gu, X. Wang, and G. Zhang. Quantum higher order singular value decomposition. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 1166–1171, Oct 2019.

[12] N. Hao, M. Kilmer, K. Braman, and R. Hoover. Facial recognition using tensor-tensor decompositions. *SIAM Journal on Imaging Sciences [electronic only]*, 6, 02 2013.

[13] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009.

[14] S. Hu, L. Qi, and G. Zhang. Computing the geometric measure of entanglement of multipartite pure states by means of non-negative tensors. *Physical Review A*, 93(1):012304, 2016.

[15] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire. Towards quantum machine learning with tensor networks. *Quantum Science and Technology*, 4(2):024001, 2019.

[16] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, pages 79–86, 2010.

[17] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash. q-means: A quantum algorithm for unsupervised machine learning. In *Advances in Neural Information Processing Systems*, pages 4134–4144, 2019.

[18] I. Kerenidis and A. Prakash. Quantum Recommendation Systems. In C. H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:21, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[19] M. E. Kilmer and C. D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.

[20] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[21] C. Liu, J. Zhou, and K. He. Image compression based on truncated hosvd. In *2009 International Conference on Information Engineering and Computer Science*, pages 1–4. IEEE, 2009.

[22] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631, 2014.

[23] G. Long. General quantum interference principle and duality computer. *Communications in Theoretical Physics*, 45(5):825, 2006.

[24] G. Long. Duality quantum computing and duality quantum information processing. *International Journal of Theoretical Physics*, 50(4):1305–1318, 2011.

[25] Y. Ma, Y. Wang, and V. Tresp. Quantum machine learning algorithm for knowledge graphs. *arXiv preprint arXiv:2001.01077*, 2020.

[26] C. D. Martin, R. Shafer, and B. LaRue. An order-p tensor factorization with applications in imaging. *SIAM Journal on Scientific Computing*, 35(1):A474–A490, 2013.

[27] M. A. Nielsen and I. Chuang. Quantum computation and quantum information, 2002.

[28] R. Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.

[29] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[30] J. Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.

[31] L. Qi, H. Chen, and Y. Chen. *Tensor Eigenvalues and Their Applications*, volume 39. Springer, 2018.

[32] L. Qi and Z. Luo. *Tensor Analysis: Spectral Theory and Special Tensors*, volume 151. Siam, 2017.

[33] L. Qi, G. Zhang, D. Braun, F. Bohnet-Waldraff, and O. Giraud. Regularly decomposable tensors and classical spin states. *Communications in Mathematical Sciences*, 2017.

[34] L. Qi, G. Zhang, and G. Ni. How entangled can a multi-party system possibly be? *Physics Letters A*, 382(22):1465–1471, 2018.

[35] D. Rafailidis and A. Nanopoulos. Modeling users preference dynamics and side information in recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(6):782–792, 2015.

[36] P. Rebentrost, M. Mohseni, and S. Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(13):130503, 2014.

[37] P. Rebentrost, M. Schuld, L. Wossnig, F. Petruccione, and S. Lloyd. Quantum gradient descent and Newton's method for constrained polynomial optimization. *New Journal of Physics*, 21(7):073023, 2019.

[38] P. Rebentrost, A. Steffens, I. Marvian, and S. Lloyd. Quantum singular-value decomposition of nonsparse low-rank matrices. *Physical Review A*, 97(1):012327, 2018.

[39] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 727–736, 2009.

[40] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pages 81–90, 2010.

[41] S. M. Ross. *Introduction to Probability Models, ISE*. Academic press, 2006.

[42] R. Sarma and R. Jain. Quantum gate implementation of a novel reversible half adder and subtractor circuit. In *2018 International Conference on Intelligent Circuits and Systems (ICICS)*, pages 72–76. IEEE, 2018.

[43] C. Shao. From linear combination of quantum states to grover's searching algorithm. *arXiv preprint arXiv:1807.09693*, 2018.

[44] C. Shao, Y. Li, and H. Li. Quantum algorithm design: Techniques and applications. *Journal of Systems Science and Complexity*, 32(1):375–452, 2019.

[45] G. Song, M. K. Ng, and X. Zhang. Robust tensor completion using transformed tensor singular value decomposition. *Numerical Linear Algebra with Applications*, 27(3):e2299, 2020.

[46] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, pages 43–50, 2008.

[47] E. Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 217–228, 2019.

[48] M. Teixeira and D. Rodriguez. A class of fast cyclic convolution algorithms based on block pseudocirculants. *IEEE Signal Processing Letters*, 2(5):92–94, 1995.

[49] H. Thapliyal and N. Ranganathan. Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 6(4):1–31, 2010.

[50] H. Thapliyal and N. Ranganathan. A new design of the reversible subtractor circuit. In *2011 11th IEEE International Conference on Nanotechnology*, pages 1430–1435. IEEE, 2011.

[51] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[52] C. Wang and L. Wossnig. A quantum algorithm for simulating non-sparse hamiltonians. *arXiv preprint arXiv:1803.08273*, 2018.

[53] M. Wu, S. He, Y. Zhang, J. Chen, Y. Sun, Y.-Y. Liu, J. Zhang, and H. V. Poor. A tensor-based framework for studying eigenvector multicentrality in multilayer networks. *Proceedings of the National Academy of Sciences*, 116(31):15407–15413, 2019.

[54] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 211–222. SIAM, 2010.

[55] G. Zhang. Dynamical analysis of quantum linear systems driven by multi-channel multi-photon states. *Automatica*, 83:186–198, 2017.

[56] J. Zhang, A. K. Saibaba, M. E. Kilmer, and S. Aeron. A randomized tensor singular value decomposition based on the t-product. *Numerical Linear Algebra with Applications*, 25(5):e2179, 2018.

[57] M. Zhang, G. Ni, and G. Zhang. Iterative methods for computing U-eigenvalues of non-symmetric complex tensors with application in quantum entanglement. *Computational Optimization and Applications*, 75:779–798, 2020.

[58] Z. Zhang and S. Aeron. Exact tensor completion using t-svd. *IEEE Transactions on Signal Processing*, 65(6):1511–1526, 2016.

[59] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3842–3849, 2014.

[60] P. Zhou, C. Lu, Z. Lin, and C. Zhang. Tensor factorization for low-rank tensor completion. *IEEE Transactions on Image Processing*, 27(3):1152–1163, 2017.

[61] X.-Q. Zhou, T. C. Ralph, P. Kalasuwan, M. Zhang, A. Peruzzo, B. P. Lanyon, and J. L. O'Brien. Adding control to arbitrary unknown quantum operations. *Nature Communications*, 2(1):1–8, 2011.