Journal of Mathematical Imaging and Vision (2020) 62:560–584
https://doi.org/10.1007/s10851-020-00946-9

Generalized Visual Information Analysis Via Tensorial Algebra

Liang Liao[1] · **Stephen John Maybank**[2]

Received: 19 April 2019 / Accepted: 28 January 2020 / Published online: 12 February 2020
© The Author(s) 2020

Abstract
High-order data are modeled using matrices whose entries are numerical arrays of a fixed size. These arrays, called t-scalars, form a commutative ring under the convolution product. Matrices with elements in the ring of t-scalars are referred to as t-matrices. The t-matrices can be scaled, added and multiplied in the usual way. There are t-matrix generalizations of positive matrices, orthogonal matrices and Hermitian symmetric matrices. With the t-matrix model, it is possible to generalize many well-known matrix algorithms. In particular, the t-matrices are used to generalize the singular value decomposition (SVD), high-order SVD (HOSVD), principal component analysis (PCA), two-dimensional PCA (2DPCA) and Grassmannian component analysis (GCA). The generalized t-matrix algorithms, namely TSVD, THOSVD, TPCA, T2DPCA and TGCA, are applied to low-rank approximation, reconstruction and supervised classification of images. Experiments show that the t-matrix algorithms compare favorably with standard matrix algorithms.

Keywords Commutative ring · Grassmannian manifold · Image classification · Tensor singular value decomposition · Tensors

1 Introduction

In data analysis, machine learning and computer vision, the data are often given in the form of multi-dimensional arrays of numbers. For example, an RGB image has three dimensions, namely two for the pixel array and a third dimension for the values of the pixels. An RGB image is said to be an array of order three. Alternatively, the RGB image is said to have three modes or to be three-way. A video sequence of images is of order four, with two dimensions for the pixel array, one dimension for time and a fourth dimension for the pixel values.

One way of analyzing multi-dimensional data is to remove the array structure by flattening, to obtain a vector. A set of vectors obtained in this way can be analyzed using standard matrix-vector algorithms such as the singular value decomposition (SVD) and principal component analysis (PCA). An alternative to flattening is to use algorithms that preserve the multi-dimensional structure. In these algorithms, the elements of matrices and vectors are entire arrays rather than real numbers in $\mathbb{R}$ or complex numbers in $\mathbb{C}$. Multi-dimensional arrays with the same dimensions can be added in the usual way, but there is no definition of multiplication which satisfies the requirements for a field such as $\mathbb{R}$ or $\mathbb{C}$. However, multiplication based on the convolution product has many but not all of the properties of a field. Convolution multiplication differs from the multiplication in a field in that many elements have no multiplicative inverse. The multi-dimensional arrays with given dimensions form a commutative ring under the convolution product. The elements of this ring are referred to as t-scalars.

An application of the Fourier transform shows that each ring of t-scalars under the convolution product is isomorphic to a ring of arrays in which the Hadamard product defines the multiplication. In effect, the ring obtained by applying the

Liang Liao and Stephen John Maybank contributed equally to this work.

This work was supported by the National Natural Science Foundation of China (No. U1404607) and the High-end Foreign Experts Program (No. GDW20186300351) of State Administration of Foreign Experts Affairs.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10851-020-00946-9) contains supplementary material, which is available to authorized users.

✉ Liang Liao
liaoliangis@126.com

Stephen John Maybank
sjmaybank@dcs.bbk.ac.uk

1 School of Electronics and Information, Zhongyuan University of Technology, Zhengzhou 450000, People's Republic of China

2 Birkbeck College, University of London, London WC1E7HX, UK

Fourier transform splits into a product of copies of $\mathbb{C}$. It is this splitting which allows the construction of new algorithms for analyzing tensorial data without flattening. The so-called t-matrices with t-scalar entries have many of the properties of matrices with elements in $\mathbb{R}$ or $\mathbb{C}$. In particular, t-matrices can be scaled, added and multiplied. There is an additive identity and a multiplicative identity. The determinant of a t-matrix is defined, and a given t-matrix is invertible if and only if it has an invertible determinant. The t-matrices include generalizations of positive matrices, orthogonal matrices and symmetric matrices.

A tensorial version, TSVD, of the SVD is described in [18] and [41]. The TSVD expresses a t-matrix as the product of three t-matrices, of which two are generalizations of the orthogonal matrices and one is a diagonal matrix with positive t-scalars on the diagonal. The TSVD is used to define tensorial versions of principal component analysis (PCA) and two-dimensional PCA (2DPCA). A tensorial version of Grassmannian component analysis is also defined. These tensorial algorithms are tested by experiments that include low-rank approximations to tensors, reconstruction of tensors and terrain classification using hyperspectral images. The different algorithms are compared using the peak signal-to-noise ratio and Cohen's kappa.

The t-scalars are described in Sect. 2, and the t-matrices are described in Sect. 3. The TSVD is described in Sect. 4. A tensorial version of principal component analysis (TPCA) is obtained from the TSVD in Sect. 5 and then generalized to tensorial two-dimensional PCA (T2DPCA). A tensorial version of Grassmannian components analysis is also defined. The tensorial algorithms are tested experimentally in Sect. 6. Some concluding remarks are made in Sect. 7.

1.1 Related Work

A tensor of order two or more can be simplified using the so-called N-mode singular value decomposition (SVD). The three-mode case is described by Tucker in [30]. The multimodal case is discussed in detail by De Lathauwer et al. [6]. Each mode of the tensor has an associated set of vectors, each one of which is obtained by varying the index for the given mode while keeping the indices of the other modes fixed. In the N-mode SVD, an orthonormal basis is obtained for the space spanned by these vectors. In the two-mode case, the result is the usual SVD. The resulting decomposition of a tensor is referred to as the higher-order SVD (HOSVD). Surveys of tensor decompositions can be found in Kolda and Bader [19] and Sidiropoulos et al. [27]. De Lathauwer et al. [6] describe a higher-order eigenvalue decomposition. Vasilescu and Terzopoulos [34] use the N-mode SVD to simplify a fifth-order tensor constructed from face images taken under varying conditions and with varying expressions. A tensor version of the singular value decomposition is described in [18,41], and [17].

He et al. [15] sample a hyperspectral data cube to yield tensors of order three of which two orders are for the pixel array and one order is for the hyperspectral bands. A training set of samples is used to produce a dictionary for sparse classification. Lu et al. [23] use N-mode analysis to obtain projections of tensors to a lower-dimensional space. The resulting multilinear PCA is applied to the classification of gait images. Vannieuwenhoven et al. [32] describe a new method for truncating the higher-order SVD, to obtain low-rank multilinear approximations to tensors. The method is tested on the classification of handwritten digits and the compression of a database of face images.

Many authors have studied algebras of matrices in which the elements are tensors of order one, equipped with a convolution multiplication, under which they form a commutative ring R with a multiplicative identity. In particular, Gleich et al. [11] describe the generalized eigenvalues and eigenvectors of matrices with elements in R and show how the standard power method for finding an eigenvector and the standard Arnoldi method for constructing an orthogonal basis for a Krylov subspace can both be generalized. Braman [3] shows that the t-vectors with a given dimension form a free module over R. Kilmer and Martin [18] show that many of the properties and structures of canonical matrices and vectors can be generalized. Their examples include transposition, orthogonality and the singular value decomposition (SVD). The tensor SVD is used to compress tensors. A tensor-based method for image de-blurring is also described. Kilmer et al. [17] generalize the inner product of two vectors, suggest a notion of the angle between two vectors with elements in R and define a notion of orthogonality for two vectors. A generalization of the Gram–Schmidt method for generating an orthonormal set of vectors is also described in [17].

Zhang et al. [41] use the tensor SVD to store video sequences efficiently and also to fill in missing entries in video sequences. Zhang et al. [39] use a randomized version of the tensor SVD to produce low-rank approximations to matrices. Ren et al. [28] define a tensor version of principal component analysis and use it to extract features from hyperspectral images. The features are classified using standard methods such as support vector machines and nearest neighbors. Liao et al. [20] generalize a sparse representation classifier to tensor data and apply the generalized classifier to image data such as numerals and faces. Chen et al. [4] use a four-dimensional HOSVD to detect changes in a time sequence of hyperspectral images. The K-means clustering algorithm is used to classify the pixel values as changed or unchanged. Fan et al. [8] model a hyperspectral image as the sum of an ideal image, a sparse noise term and a Gaussian noise term. A product of two low-rank tensors models the ideal image. The low-rank tensors are estimated by minimiz-

ing a penalty function obtained by adding the squared errors in a fit of the hyperspectral image to penalty terms for the sparse noise and the sizes of the two low-rank tensors. Lu et al. [22] approximate a third-order tensor using the sum of a low-rank tensor and a sparse tensor. Under suitable conditions, the low-rank tensor and the sparse tensor are recovered exactly.

2 T-Scalars

The notations for t-scalars are summarized in Sect. 2.1. Basic definitions are given in Sect. 2.2. The Fourier transform of a t-scalar is defined in Sect. 2.3. Properties of t-scalars and the Fourier transform of a t-scalar are described in Sect. 2.4. A generalization of the t-scalars is described in Sect. 3.5.

2.1 Notations and Preliminaries

An array of order N over the complex numbers $\mathbb{C}$ is an element of the set C defined by $C \equiv \mathbb{C}^{I_1 \times \cdots \times I_N}$, where the I_n for $1 \leq n \leq N$ are strictly positive integers. Similarly, an array of order N over the real numbers is an element of the set R defined by $R \equiv \mathbb{R}^{I_1 \times \cdots \times I_N}$. The sets R and C have the structure of commutative rings, in which the product is defined by circular convolution. The elements of C and R are referred to as t-scalars.

Elements of $\mathbb{R}$ and $\mathbb{C}$ are denoted by lower-case letters and tensorial data are denoted by upper-case letters. The t-scalars are identified using the subscript T, for example X_T. Lower-case subscripts such as i, j, α, β are indices or lists of indices.

All indices begin from 1 rather than 0. Given an array of any order N, namely $X \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ $(N \geqslant 1)$, $X_{i_1,i_2,\ldots,i_N}$ or $(X)_{i_1,i_2,\ldots,i_N}$ denote its $(i_1, i_2, \ldots, i_N)$th entry in $\mathbb{C}$. The notation X_i, or $(X)_i$, is also used, where i is a multi-index defined by $i = (i_1, \ldots, i_N)$. Let $I = (I_1, I_2, \ldots, I_N)$ and let i be a multi-index. The notation $1 \leq i \leq I$ specifies the range of values of i such that $1 \leq i_n \leq I_n$ for $1 \leq n \leq N$. It is often convenient to extend the indexing beyond the range specified by I. Let j be a general multi-index. Then, X_j is defined by $X_j = X_i$, where i is the multi-index such that each component i_n is in the range $1 \leq i_n \leq I_n$ and $i_n - j_n$ is divisible by I_n. A multi-index such as $i - j + 1$ has components $i_n - j_n + 1$ for $1 \leq n \leq N$. The sum $\sum_{i=1}^{I}(\cdot)$ is an abbreviation for $\sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N}(\cdot)$.

2.2 Definitions

The following definitions are for t-scalars in C. Similar definitions can be made for t-scalars in R.

Definition 1 T-scalar addition. Given t-scalars X_T and Y_T in C, the addition of X_T and Y_T denoted by $D_T \doteq X_T + Y_T$ is element-wise:

$$D_{T,i} = X_{T,i} + Y_{T,i}, \quad 1 \leq i \leq I. \tag{1}$$

Definition 2 T-scalar multiplication. Given t-scalars X_T and Y_T in C, their product, denoted by $D_T = X_T \circ Y_T$, is a t-scalar in C defined by the circular convolution

$$D_{T,i} = \sum_{j=1}^{I} X_{T,i-j+1} Y_{T,j}, \quad 1 \leq i \leq I. \tag{2}$$

Definitions 1 and 2 reduce to complex number addition and multiplication when $N = 1$ and $I_1 = 1$.

Definition 3 Zero t-scalar. The zero t-scalar Z_T is the array in C defined by

$$Z_{T,i} = 0, \quad 1 \leq i \leq I. \tag{3}$$

For all t-scalars X_T, $X_T + Z_T = X_T$ and $X_T \circ Z_T = Z_T$.

Definition 4 Identity t-scalar. The identity t-scalar E_T in C has the first entry equal to 1 and all other entries equal to 0, namely $E_{T,i} = 1$ if $i = (1, \ldots, 1)$ and $E_{T,i} = 0$ otherwise.

For all t-scalars $X_T \in C$, $X_T \circ E_T \equiv X_T$.

The set of t-scalars satisfies the axioms of a commutative ring with Z_T as an additive identity and E_T as a multiplicative identity. This ring of t-scalars is denoted by $(C, +, \circ)$. The ring $(C, +, \circ)$ is a generalization of the field $(\mathbb{C}, +, \cdot)$ of complex numbers. If the t-scalars are restricted to have real number elements, then the ring $(R, +, \circ)$ is obtained.

2.3 Fourier Transform of a T-Scalar

Let ζ_n be a primitive I_nth root of unity, for example,

$$\zeta_n = \exp\left(2\pi\sqrt{-1}/I_n\right), \quad 1 \leq n \leq N.$$

Let $\overline{\zeta}_n$ be the complex conjugate of ζ_n, and let X_T be a t-scalar in the ring C. The Fourier transform $F(X_T)$ of X_T is defined by

$$F(X_T)_i = \sum_{j=1}^{I} X_{T,j} \cdot \zeta_1^{(i_1-1)(j_1-1)} \ldots \zeta_N^{(i_N-1)(j_N-1)}$$

for all indices $1 \leq i \leq I$.

The inverse of the Fourier transform is defined by

$$X_{T,i} = \frac{\sum_{j=1}^{I} F(X_T)_j \cdot \overline{\zeta}_1^{(i_1-1)(j_1-1)} \ldots \overline{\zeta}_N^{(i_N-1)(j_N-1)}}{I_1 \ldots I_N}$$

for all indices $1 \leq i \leq I$.

Given t-scalars $X_T \in C$ and $Y_T \in C$ and their t-scalar product $D_T = X_T \circ Y_T$, it follows that

$$F(D_T) = F(X_T) * F(Y_T), \tag{4}$$

where $*$ denotes the Hadamard product in C. Equation (4) is an extension of the convolution theorem [2]. The equation can be equivalently rewritten as

$$F(D_T)_i = F(X_T)_i \cdot F(Y_T)_i, \quad 1 \le i \le I, \tag{5}$$

where $\cdot$ is multiplication in $\mathbb{C}$.

An equivalent definition of the Fourier transform of a high-order array in the form of multi-mode tensor multiplication and a diagram of the multiplication of two t-scalars, computed in the Fourier domain, is given in a supplementary file.

It is not difficult to prove that C is a commutative ring, $(C, +, *)$, under the Hadamard product. The Fourier transform is a ring isomorphism from $(C, +, \circ)$ to $(C, +, *)$. The identity element of $(C, +, *)$ is $J_T = F(E_T)$. All the entries of J_T are equal to 1.

2.4 Properties of T-Scalars

The invertible t-scalars are defined as follows.

Definition 5 Invertible t-scalar: Given a t-scalar X_T, if there exists a t-scalar Y_T satisfying $X_T \circ Y_T = E_T$, then X_T is said to be invertible. The t-scalar Y_T is the inverse of X_T and denoted by $Y_T \doteq X_T^{-1} \doteq E_T / X_T$.

The zero t-scalar Z_T is non-invertible. In addition, there are an infinite number of t-scalars that are non-invertible. For example, given a t-scalar $X_T \in C$, if the entries of X_T are all equal, then X_T is non-invertible. The existence of more than one non-invertible element shows that C is not a field.

Definition 6 Scalar multiplication of a t-scalar. Given a scalar $\lambda \in \mathbb{C}$ and a t-scalar $X_T \in C$, their product, denoted by $Y_T = \lambda \cdot X_T \equiv X_T \cdot \lambda$, is the t-scalar given by

$$Y_{T,i} = \lambda \cdot X_{T,i}, \quad 1 \le i \le I. \tag{6}$$

It can be shown that the set of t-scalars is a vector space over $\mathbb{C}$.

The following definition of the conjugate of a t-scalar generalizes the conjugate of a complex number.

Definition 7 Conjugate of a t-scalar. Given a t-scalar X_T in C, its conjugate, denoted by $\mathrm{conj}(X_T)$, is the t-scalar in C such that

$$\mathrm{conj}(X_T)_i = \overline{X_{T,2-i}}\ , \quad 1 \le i \le I, \tag{7}$$

where $\overline{X_{T,2-i}}$ is the complex conjugate of $X_{T,2-i}$ in $\mathbb{C}$.

The conjugate of a t-scalar reduces to the conjugate of a complex number when $N = 1$, $I_1 = 1$. The relationship of $\mathrm{conj}(X_T)$ and X_T is much clearer if they are mapped onto the Fourier domain—each entry of $F(\mathrm{conj}(X_T))$ is the complex conjugate of the corresponding entry of $F(X_T)$, namely

$$F(\mathrm{conj}(X_T))_i = \overline{F(X_T)_i}\ , \quad 1 \le i \le I. \tag{8}$$

It follows from Eq. (7) that $\mathrm{conj}(\mathrm{conj}(X_T)) = X_T$ for any $X_T \in C$.

Definition 8 Self-conjugate t-scalar: Given a t-scalar $X_T \in C$, if $X_T = \mathrm{conj}(X_T)$, then X_T is said to be a self-conjugate t-scalar.

If X_T is self-conjugate, then

$$\overline{F(X_T)_i} = F(\mathrm{conj}(X_T))_i = F(X_T)_i \in \mathbb{C}\ , 1 \le i \le I. \tag{9}$$

It follows from Eq. (9) that X_T is self-conjugate if and only if all the elements of $F(X_T)$ are real numbers.

The t-scalars Z_T and E_T are both self-conjugate. Furthermore, the self-conjugate t-scalars form a ring denoted by C^{sc}. This ring is a subring of C.

Given any t-scalar $X_T \in C$, let $\Re(X_T)$ and $\Im(X_T)$ be defined by

$$\Re(X_T) = 2^{-1}(X_T + \mathrm{conj}(X_T)), \tag{10}$$

$$\Im(X_T) = \left(2\sqrt{-1}\right)^{-1}(X_T - \mathrm{conj}(X_T)). \tag{11}$$

It follows from Eq. (9) that $\Re(X_T)$ and $\Im(X_T)$ are self-conjugate. The t-scalars $X_T \in C$ and $\mathrm{conj}(X_T) \in C$ can be expressed in the form

$$X_T = \Re(X_T) + \sqrt{-1}\Im(X_T), \tag{12}$$

$$\mathrm{conj}(X_T) = \Re(X_T) - \sqrt{-1}\Im(X_T). \tag{13}$$

In an analogy with the real and imaginary parts of a complex number, $\Re(X_T)$ is called the real part of X_T and $\Im(X_T)$ is called the imaginary part of X_T.

Given two t-scalars X_T and Y_T, Eq. (14) holds true and is backward compatible with the corresponding equations for complex numbers.

$$\begin{aligned} X_T + Y_T &\equiv \left(\Re(X_T) + \Re(Y_T)\right) + \sqrt{-1} \cdot \left(\Im(X_T) + \Im(Y_T)\right) \\ X_T \circ Y_T &\equiv \left(\Re(X_T) \circ \Re(Y_T) - \Im(X_T) \circ \Im(Y_T)\right) \\ &\quad + \sqrt{-1}\left(\Im(X_T) \circ \Re(Y_T) + \Re(X_T) \circ \Im(Y_T)\right) \\ \mathrm{conj}(X_T) \circ X_T &\equiv X_T \circ \mathrm{conj}(X_T) \equiv \Re(X_T)^2 + \Im(X_T)^2 \end{aligned} \tag{14}$$

Definition 9 Nonnegative t-scalar: The t-scalar X_T is said to be nonnegative if there exists a self-conjugate t-scalar Y_T such that $X_T = Y_T \circ Y_T \doteq Y_T^2$.

If a t-scalar X_T is nonnegative, it is also self-conjugate, because the multiplication of any two self-conjugate t-scalars is also a self-conjugate t-scalar. Thus, both Z_T and E_T are nonnegative, since Z_T and E_T are self-conjugate t-scalars and satisfy $Z_T = Z_T^2$ and $E_T = E_T^2$. Furthermore, for all $X_T \in C$, the ring element $\Re(X_T)^2 + \Im(X_T)^2$ is nonnegative.

The set S^{nonneg} of nonnegative t-scalars is closed under the t-scalar addition and multiplication. Since a nonnegative t-scalar is also a self-conjugate t-scalar, $S^{\text{nonneg}} \subset C^{sc} \subset C$.

Theorem 1 *For all t-scalars $X_T \in S^{\text{nonneg}}$, there exists a unique t-scalar $S_T \in S^{\text{nonneg}}$ satisfying $X_T = S_T \circ S_T \doteq S_T^2$. We call the nonnegative t-scalar S_T the arithmetic square root of the nonnegative t-scalar X_T and denote it by*

$$S_T \doteq \sqrt{X_T} \doteq X_T^{1/2}. \tag{15}$$

Proof Let $X_T = Y_T \circ Y_T$, such that Y_T is self-conjugate. On applying the Fourier transform, it follows that

$$F(X_T)_i = F(Y_T)_i^2 \geq 0, \quad 1 \leq i \leq I.$$

Let S_T be defined such that

$$F(S_T)_i = (F(X_T)_i)^{1/2}, \quad 1 \leq i \leq I,$$

where the nonnegative square root is chosen for each value of i. The Fourier components $F(S_T)_i$ are real-valued; thus, S_T is self-conjugate. The equation $X_T = S_T \circ S_T$ holds because the Fourier transform is injective. □

Definition 10 A nonnegative t-scalar that is invertible under multiplication is called a positive t-scalar. The set of positive t-scalars is denoted by S^{pos}.

The following inclusions are strict, $S^{\text{pos}} \subset S^{\text{nonneg}} \subset C^{\text{sc}} \subset C$. The inverse and the arithmetic square root of a positive t-scalar are positive.

The absolute t-value $r(X_T)$ of X_T is defined by

$$r(X_T) = \sqrt{\Re(X_T)^2 + \Im(X_T)^2}. \tag{16}$$

The t-scalars $\Re(X_T)$ and $\Im(X_T)$ are both self-conjugate; therefore, $\Re(X_T)^2$ and $\Im(X_T)^2$ are both nonnegative. The sum $\Re(X_T)^2 + \Im(X_T)^2$ is nonnegative, and it has a nonnegative arithmetical square root, namely $r(X_T)$.

If $r(X_T)$ is invertible, then let $\phi(X_T)$ be defined by

$$\phi(X_T) \doteq r(X_T)^{-1} \circ X_T. \tag{17}$$

The ring element $\phi(X_T)$ is a generalized angle. The order 1 version of $\phi(X_T)$ is obtained by Gleich et al. [11]. Equation (17) generalizes the polar form of a complex number. It can be shown that

$$\phi(X_T) \circ \text{conj}(\phi(X_T)) = E_T.$$

The absolute t-value $r(X_T)$ is used in Sect. 3 to define a generalization of the Frobenius norm for t-matrices.

3 Matrices with T-Scalar Elements

It is shown that t-matrices, i.e., matrices with elements in the rings C or R, are in many ways analogous to matrices with elements in $\mathbb{C}$ or $\mathbb{R}$.

3.1 Indexing

The t-matrices are order-two arrays of t-scalars. Since the t-scalars are arrays of complex numbers, it is convenient to organize t-matrices as hierarchical arrays of complex numbers.

Let X_{TM} be a t-matrix with D_1 rows and D_2 columns. Then, X_{TM} is an element of $C^{D_1 \times D_2}$. The (α, β) entry of X_{TM} is the element of C denoted by $X_{\text{TM},\alpha,\beta}$ for $1 \leq \alpha \leq D_1$ and $1 \leq \beta \leq D_2$. Let i be a multi-index for elements of C. Then, $X_{\text{TM},i,\alpha,\beta}$ is the element of $\mathbb{C}$ given as the ith entry of the ring element $X_{\text{TM},\alpha,\beta}$.

The t-matrix X_{TM} can be interpreted as an element in $\mathbb{C}^{I_1 \times \cdots \times I_N \times D_1 \times D_2}$, or alternatively it can be interpreted as an element in $\mathbb{C}^{D_1 \times D_2 \times I_1 \times \cdots \times I_N}$. The only thing needed to switch from one data structure to the other is a permutation of indices. The data structure $\mathbb{C}^{I_1 \times \cdots \times I_N \times D_1 \times D_2}$ is chosen unless otherwise indicated.

3.2 Properties of t-Matrices

(1) **T-matrix addition**: Given any t-matrices $A_{\text{TM}} \in C^{D_1 \times D_2}$ and $B_{\text{TM}} \in C^{D_1 \times D_2}$, the addition, denoted by $C_{\text{TM}} \doteq A_{\text{TM}} + B_{\text{TM}} \in C^{D_1 \times D_2}$, is entry-wise, such that $C_{\text{TM},\alpha,\beta} = A_{\text{TM},\alpha,\beta} + B_{\text{TM},\alpha,\beta}$, for $1 \leq \alpha \leq D_1$ and $1 \leq \beta \leq D_2$.

(2) **T-matrix multiplication**: Given any t-matrices $A_{\text{TM}} \in C^{D_1 \times Q}$ and $B_{\text{TM}} \in C^{Q \times D_2}$, their product, denoted by $C_{\text{TM}} \doteq A_{\text{TM}} \circ B_{\text{TM}}$, is the t-matrix in $C^{D_1 \times D_2}$ defined by

$$C_{\text{TM},\alpha,\beta} = \sum_{\gamma=1}^{Q} A_{\text{TM},\alpha,\gamma} \circ B_{\text{TM},\gamma,\beta}$$

for all indices $1 \leq \alpha \leq D_1$, $1 \leq \beta \leq D_2$.
An example of t-matrix multiplication $C_{\text{TM}} = A_{\text{TM}} \circ B_{\text{TM}} \in C^{2\times 1} \equiv \mathbb{C}^{3\times 3\times 2\times 1}$ where $A_{\text{TM}} \in C^{2\times 2} \equiv \mathbb{C}^{3\times 3\times 2\times 2}$ and $B_{\text{TM}} \in C^{2\times 1} \equiv \mathbb{C}^{3\times 3\times 2\times 1}$ is given in a supplementary file.

(3) **Identity t-matrix**: The identity t-matrix is the diagonal t-matrix, in which each diagonal entry is equal to the identity t-scalar E_T in Definition 4. The $D \times D$ identity t-matrix is denoted by $I_{\mathrm{TM}}^{(D)} \doteq \mathrm{diag}(\underbrace{E_T, \cdots, E_T}_{D})$.

Given any $X_{\mathrm{TM}} \in C^{D_1 \times D_2}$, it follows that $I_{\mathrm{TM}}^{(D_1)} \circ X_{\mathrm{TM}} = X_{\mathrm{TM}} \circ I_{\mathrm{TM}}^{(D_2)} = X_{\mathrm{TM}}$. The identity t-matrix $I_{\mathrm{TM}}^{(D)}$ is also denoted by I_{TM} if the value of D can be inferred from context.

(4) **Scalar multiplication**: Given any $A_{\mathrm{TM}} \in C^{D_1 \times D_2}$ and $\lambda \in \mathbb{C}$, their multiplication, denoted by $B_{\mathrm{TM}} \doteq \lambda \cdot A_{\mathrm{TM}}$, is the t-matrix in $C^{D_1 \times D_2}$ defined by

$$B_{\mathrm{TM},\alpha,\beta} = \lambda \cdot A_{\mathrm{TM},\alpha,\beta}, \quad 1 \le \alpha \le D_1, 1 \le \beta \le D_2,$$

where the products with λ are computed as in Definition 6.

(5) **T-scalar multiplication**: Given any $A_{\mathrm{TM}} \in C^{D_1 \times D_2}$ and $\lambda_T \in C$, their product, denoted by $B_{\mathrm{TM}} \doteq \lambda_T \circ A_{\mathrm{TM}}$, is the t-matrix in $C^{D_1 \times D_2}$ defined by

$$B_{\mathrm{TM},\alpha,\beta} = \lambda_T \circ A_{\mathrm{TM},\alpha,\beta}, \quad 1 \le \alpha \le D_1, 1 \le \beta \le D_2.$$

(6) **Conjugate transpose of a t-matrix**: Given any t-matrix $X_{\mathrm{TM}} \in C^{D_1 \times D_2}$, its conjugate transpose, denoted by $X_{\mathrm{TM}}^{\mathcal{H}}$, is the t-matrix in $C^{D_2 \times D_1}$ given by

$$X_{\mathrm{TM},\beta,\alpha}^{\mathcal{H}} = \mathrm{conj}(X_{\mathrm{TM},\alpha,\beta}) \in C, \quad 1 \le \alpha \le D_1, 1 \le \beta \le D_2.$$

A square matrix U_{TM} is said to be orthogonal if $U_{\mathrm{TM}}^{\mathcal{H}}$ is the inverse t-matrix of U_{TM}, i.e., $U_{\mathrm{TM}}^{\mathcal{H}} \circ U_{\mathrm{TM}} = U_{\mathrm{TM}} \circ U_{\mathrm{TM}}^{\mathcal{H}} = I_{\mathrm{TM}}$. The Fourier transform F is extended to t-matrices element-wise, i.e., $F(X_{\mathrm{TM}})$ is the $D_1 \times D_2$ t-matrix defined by

$$F(X_{\mathrm{TM}})_{\alpha,\beta} = F(X_{\mathrm{TM},\alpha,\beta}) \quad , \tag{18}$$

for all indices $1 \le \alpha \le D_1$ and $1 \le \beta \le D_2$.
It is not difficult to prove that

$$F(X_{\mathrm{TM}}^{\mathcal{H}})_{i,\beta,\alpha} = \overline{F(X_{\mathrm{TM}})_{i,\alpha,\beta}} \in \mathbb{C} \,,$$

for all indices $1 \le i \le I, 1 \le \alpha \le D_1, 1 \le \beta \le D_2$.

(7) **T-vector dot product and the Frobenius norm**: Given any two t-vectors (i.e., two t-matrices, each having only one column) X_{TV} and Y_{TV} of the same length D, their dot product is the t-scalar defined by

$$\langle X_{\mathrm{TV}}, Y_{\mathrm{TV}} \rangle \doteq \sum_{\alpha=1}^{D} \mathrm{conj}(X_{\mathrm{TV},\alpha}) \circ Y_{\mathrm{TV},\alpha} \quad .$$

If $\langle X_{\mathrm{TV}}, Y_{\mathrm{TV}} \rangle = Z_T$, then X_{TV} and Y_{TV} are said to be orthogonal. The nonnegative t-scalar $\sqrt{\langle X_{\mathrm{TV}}, X_{\mathrm{TV}} \rangle}$ is called the generalized norm of X_{TV} and denoted by

$$\|X_{\mathrm{TV}}\|_F \doteq \sqrt{\langle X_{\mathrm{TV}}, X_{\mathrm{TV}} \rangle} \equiv \left(\sum_{\alpha=1}^{D} r(X_{\mathrm{TV},\alpha})^2 \right)^{1/2} , \tag{19}$$

where $r(\cdot)$ is the absolute t-value as defined by Eq. (16). The generalized Frobenius norm of a $D_1 \times D_2$ t-matrix W_{TM} is defined by

$$\|W_{\mathrm{TM}}\|_F \doteq \left(\sum_{\alpha=1}^{D_1} \sum_{\beta=1}^{D_2} r(W_{\mathrm{TM},\alpha,\beta})^2 \right)^{1/2} . \tag{20}$$

In order to have a mechanism to connect t-matrices with matrices with elements in $\mathbb{C}$ or $\mathbb{R}$, the slices of a t-matrix are defined as follows.

(8) **Slice of a t-matrix**: Any t-matrix $X_{\mathrm{TM}} \in C^{D_1 \times D_2}$, organized as an array in $\mathbb{C}^{I_1 \times \cdots \times I_N \times D_1 \times D_2}$, can be sliced into $\prod_{n=1}^{N} I_n$ matrices in $\mathbb{C}^{D_1 \times D_2}$, indexed by the multi-index i. Let $X_{\mathrm{TM}}(i) \in \mathbb{C}^{D_1 \times D_2}$ be the ith slice. The entries of $X_{\mathrm{TM}}(i)$ are complex numbers in $\mathbb{C}$ given by

$$(X_{\mathrm{TM}}(i))_{\alpha,\beta} = X_{\mathrm{TM},i,\alpha,\beta} \in \mathbb{C}$$

for all indices $1 \le i \le I, 1 \le \alpha \le D_1, 1 \le \beta \le D_2$.
The t-vectors with a given dimension form an algebraic structure called a module over the ring C [16]. Modules are generalizations of vector spaces [17]. The t-vector whose entries are all equal to Z_T is denoted by Z_{TV} and called the zero t-vector. The next step is to define what is meant by a set of linearly independent t-vectors and what is meant by a full column rank t-matrix.

(9) **Linear independence in t-vector module**: The t-vectors in a subset $\{X_{\mathrm{TV},1}, X_{\mathrm{TV},2}, \ldots, X_{\mathrm{TV},K}\}$ of a t-vector module are said to be linearly independent if the equation $\sum_{k=1}^{K} \lambda_{T,k} \circ X_{\mathrm{TV},k} = Z_{\mathrm{TV}}$ holds true if and only if $\lambda_{T,k} = Z_T$, $1 \le k \le K$.
If the t-vectors $X_{\mathrm{TV},i}$, $1 \le i \le K$, are linearly independent, then they are said to have a rank of K. If the t-vectors $Y_{\mathrm{TV},i}$ for $1 \le i \le K'$ are linearly independent and span the same sub-module as the $X_{\mathrm{TM},i}$, then $K = K'$. For further information, see [16].

(10) **Full column rank t-matrix**: A t-matrix is said to be of full column rank if all its column t-vectors are linearly independent.

3.3 T-Matrix Analysis Via the Fourier Transform

The Fourier transform of the t-matrix $X_{\mathrm{TM}} \in C^{D_1 \times D_2}$ is the t-matrix in $C^{D_1 \times D_2}$ given by Eq. (18).

Many t-matrix computations can be carried out efficiently using the Fourier transform. For example, any multiplication $C_{\mathrm{TM}} = X_{\mathrm{TM}} \circ Y_{\mathrm{TM}} \in \mathbb{C}^{I_1 \times \cdots \times I_N \times D_1 \times D_2}$, where $X_{\mathrm{TM}} \in \mathbb{C}^{I_1 \times \cdots \times I_N \times D_1 \times Q}$, $Y_{\mathrm{TM}} \in \mathbb{C}^{I_1 \times \cdots \times I_N \times Q \times D_2}$, can be decomposed to $\prod_{n=1}^{N} I_n$ matrix multiplications over the complex numbers, namely

$$F(C_{\mathrm{TM}})_{i,\alpha,\beta} = \sum\nolimits_{\gamma=1}^{Q} F(X_{\mathrm{TM}})_{i,\alpha,\gamma} \cdot F(Y_{\mathrm{TM}})_{i,\gamma,\beta} \tag{21}$$

for all indices $1 \le i \le I, 1 \le \alpha \le D_1, 1 \le \beta \le D_2$.

The conjugate transpose $X_{\mathrm{TM}}^{\mathcal{H}} \in \mathbb{C}^{I_1 \times \cdots \times I_N \times D_2 \times D_1}$ of any t-matrix $X_{\mathrm{TM}} \in \mathbb{C}^{I_1 \times \cdots \times I_N \times D_1 \times D_2}$ can be decomposed to $\prod_{n=1}^{N} I_n$ canonical conjugate transposes of matrices:

$$F(X_{\mathrm{TM}}^{\mathcal{H}})_{i,\beta,\alpha} = \overline{F(X_{\mathrm{TM}})_{i,\alpha,\beta}} \tag{22}$$

for all indices $1 \le i \le I, 1 \le \alpha \le D_1, 1 \le \beta \le D_2$. Each slice of $F\left(I_{\mathrm{TM}}^{(D)}\right)$ is the canonical identity matrix with elements in $\mathbb{C}$.

The Fourier transform decomposes a t-matrix computation such as multiplication to $\prod_{n=1}^{N} I_n$ independent complex matrix computations in the Fourier domain. The ith ($1 \le i \le I$) computation involves only the ith slices of the associated t-matrices. This fact underlies an approach for speeding-up t-matrix algorithms using parallel computations. This independence of the data in the Fourier domain makes it possible to implement parallel computing using the so-called vectorization programming (also known as array programming), which is supported by many programming languages including MATLAB, R, NumPy, Julia and Fortran.

3.4 Pooling

Sometimes, it is necessary to have a pooling mechanism to transform t-scalars to scalars in $\mathbb{R}$ or $\mathbb{C}$. Given any t-scalar $X_T \in C$, its pooling result $P(X_T) \in \mathbb{C}$ is defined by

$$P(X_T) = (I_1 \ldots I_N)^{-1} \sum_{i=1}^{I} X_{T,i}. \tag{23}$$

The pooling operation for t-matrices transforms each t-scalar entry to a scalar. More formally, given any t-matrix $Y_{\mathrm{TM}} \in C^{D_1 \times D_2}$, its pooling result $P(Y_{\mathrm{TM}})$ is by definition the matrix in $\mathbb{C}^{D_1 \times D_2}$ given by

$$P(Y_{\mathrm{TM}})_{\alpha,\beta} = P(Y_{\mathrm{TM},\alpha,\beta}),\ 1 \le \alpha \le D_1, 1 \le \beta \le D_2. \tag{24}$$

The pooling of t-vectors is a special case of Eq. (24).

3.5 Generalized Tensors

Generalized tensors, called g-tensors, generalize t-matrices and canonical tensors. The generalized tensors defined in this section are used to construct the higher-order TSVD in Sect. 4.2. A g-tensor, denoted by $X_{GT} \in C^{D_1 \times D_2 \times \cdots \times D_M}$, is a generalized tensor with t-scalar entries (i.e., an order-M array of t-scalars). Its t-scalar entries are indexed by $(X_{GT})_{\alpha_1,\ldots,\alpha_M}$. Then, a generalized mode-k multiplication of X_{GT}, denoted by $M_{GT} \doteq X_{GT} \circ_k Y_{\mathrm{TM}}$ where $Y_{\mathrm{TM}} \in C^{J \times D_k}$ and $1 \le k \le M$, is a g-tensor in $C^{D_1 \times \cdots \times D_{k-1} \times J \times D_{k+1} \times \cdots \times D_M}$ defined as follows:

$$\begin{aligned}(M_{GT})_{\alpha_1,\ldots,\alpha_{k-1},\beta,\alpha_{k+1},\ldots,\alpha_M} \\ = \sum_{\alpha_k=1}^{D_k} (X_{GT})_{\alpha_1,\ldots,\alpha_{k-1},\alpha_k,\alpha_{k+1}\ldots,\alpha_M} \circ (Y_{\mathrm{TM}})_{\beta,\alpha_k}.\end{aligned} \tag{25}$$

The **generalized mode-k flattening** of a g-tensor $X_{GT} \in C^{D_1 \times D_2 \times \cdots \times D_M}$ is an (K_1, K_2)-reshaping where $K_1 = \{k\}$ and $K_2 = \{1, \ldots, M\} \setminus \{k\}$. The result is a t-matrix in $C^{D_k \times D_k^{-1} \cdot \prod_{m=1}^{M} D_m}$. Each column of the matrix is obtained by holding the indices in K_2 fixed and varying the index in K_1.

The generalized mode-k multiplication defined in Eq. (25) can also be expressed in terms of unfolded g-tensors:

$$M_{GT} = X_{GT} \circ_k Y_{\mathrm{TM}} \quad \Leftrightarrow \quad M_{GT(k)} = Y_{\mathrm{TM}} \circ X_{GT(k)},$$

where $M_{GT(k)} \in C^{J \times (D_1 \ldots D_{k-1} D_{k+1} \ldots D_M)}$ and $X_{GT(k)} \in C^{D_k \times (D_1 \ldots D_{k-1} D_{k+1} \ldots D_M)}$ are, respectively, the generalized mode-k flattening of the g-tensors M_{GT} and X_{GT}.

An example of a generalized tensor (g-tensor) $X_{GT} \in C^{2 \times 3 \times 2} \equiv \mathbb{C}^{3 \times 3 \times 2 \times 3 \times 2}$, its mode-$k$ flattening and its mode-2 multiplication with a t-matrix $Y_{\mathrm{TM}} \in C^{2 \times 3} \equiv \mathbb{C}^{3 \times 3 \times 2 \times 3}$ are given in a supplementary file.

4 Tensor Singular Value Decomposition

The singular value decomposition (SVD) is a well-known factorization of real or complex matrices [12]. It generalizes the eigendecomposition of positive semi-definite normal matrices to non-square and non-normal matrices. The SVD has a wide range of applications in data analytics, including computing the pseudo-inverse of a matrix, solving linear least square problems, low-rank approximation and linear and multilinear component analysis. A tensor version TSVD of the SVD is described in Sect. 4.1 and then applied in Sect. 4.2 to obtain a tensor version, THOSVD, of the higher-order SVD (HOSVD). Further information about the TSVD can be found in [18] and [41].

4.1 TSVD: Tensorial SVD

4.1.1 Algorithm

A tensor version, TSVD, of the singular value decomposition is described in this section and then applied in Sect. 4.2 to obtain a tensor version of the high-order SVD (HOSVD). See [41] and [18].

Given a t-matrix $X_{\mathrm{TM}} \in C^{D_1 \times D_2}$, let $Q \doteq \min(D_1, D_2)$. The TSVD of X_{TM} yields the following three t-matrices $U_{\mathrm{TM}} \in C^{D_1 \times Q}$, $S_{\mathrm{TM}} \in C^{Q \times Q}$ and $V_{\mathrm{TM}} \in C^{D_2 \times Q}$, such that

$$X_{\mathrm{TM}} = U_{\mathrm{TM}} \circ S_{\mathrm{TM}} \circ V_{\mathrm{TM}}^{\mathcal{H}}, \tag{26}$$

where $U_{\mathrm{TM}}^{\mathcal{H}} \circ U_{\mathrm{TM}} = V_{\mathrm{TM}}^{\mathcal{H}} \circ V_{\mathrm{TM}} = I_{\mathrm{TM}}^{(Q)}$, $S_{\mathrm{TM}} = \mathrm{diag}(\lambda_{T,1}, \ldots, \lambda_{T,Q})$ and $\lambda_{T,1}, \ldots, \lambda_{T,Q} \in C$ are nonnegative and satisfy $F(\lambda_{T,1})_i \geq \cdots \geq F(\lambda_{T,Q})_i \geq 0$, $1 \leq i \leq I$. The t-matrices U_{TM} and V_{TM} are generalizations of the orthogonal matrices in the SVD of a matrix with elements in $\mathbb{R}$ or $\mathbb{C}$.

Although it is possible to compute U_{TM}, S_{TM} and V_{TM} in the spatial domain, it is preferable to organize the TSVD algorithm in the Fourier domain, because of the observation in Sect. 2.3 that the Fourier transform converts the convolution product to the Hadamard product. The TSVD of X_{TM} can be decomposed into $\prod_{n=1}^{N} I_n$ SVDs of complex number matrices given by the slices of the Fourier transform $F(X_{\mathrm{TM}})$. The t-matrices U_{TM}, S_{TM} and V_{TM} in Eq. (26) are obtained in Algorithm 1.

Algorithm 1 $(U_{\mathrm{TM}}, S_{\mathrm{TM}}, V_{\mathrm{TM}}) = \mathtt{tsvd}(X_{\mathrm{TM}})$

Inputs: A t-matrix $X_{\mathrm{TM}} \in C^{D_1 \times D_2}$ as in Eq. (26) and the t-scalar dimensions I.

Outputs: The t-matrices U_{TM}, S_{TM} and V_{TM} as in Eq. (26)

1: Compute the Fourier transformation $\tilde{X}_{\mathrm{TM}} \leftarrow F(X_{\mathrm{TM}})$

2: **for all** $1 \leq i \leq I$ **do**

3: Compute the canonical SVD of $\tilde{X}_{\mathrm{TM}}(i) \in \mathbb{C}^{D_1 \times D_2}$ such that $\tilde{X}_{\mathrm{TM}}(i) = U_{\mathrm{mat}} \cdot S_{\mathrm{mat}} \cdot V_{\mathrm{mat}}^{H}$

where $U_{\mathrm{mat}} \in \mathbb{C}^{D_1 \times Q}$, $S_{\mathrm{mat}} \in \mathbb{C}^{Q \times Q}$, $V_{\mathrm{mat}} \in \mathbb{C}^{D_2 \times Q}$, $Q \doteq \min(D_1, D_2)$ and V_{mat}^{H} denotes the conjugate transpose of the complex matrix V_{mat}.

4: Assign the ith slices $\tilde{U}_{\mathrm{TM}} \in C^{D_1 \times Q}$, $\tilde{S}_{\mathrm{TM}} \in C^{Q \times Q}$ and $\tilde{V}_{\mathrm{TM}} \in C^{D_2 \times Q}$ in the Fourier domain

$$\tilde{U}_{\mathrm{TM}}(i) \leftarrow U_{\mathrm{mat}}, \quad \tilde{S}_{\mathrm{TM}}(i) \leftarrow S_{\mathrm{mat}}, \quad \tilde{V}_{\mathrm{TM}}(i) \leftarrow V_{\mathrm{mat}}.$$

5: **end for**

6: Compute the inverse transforms to obtain U_{TM}, S_{TM} and V_{TM}

$U_{\mathrm{TM}} \leftarrow F^{-1}(\tilde{U}_{\mathrm{TM}})$, $S_{\mathrm{TM}} \leftarrow F^{-1}(\tilde{S}_{\mathrm{TM}})$, $V_{\mathrm{TM}} \leftarrow F^{-1}(\tilde{V}_{\mathrm{TM}})$.

7: **return** U_{TM}, S_{TM}, V_{TM}.

If X_{TM} is defined over $\mathbb{R}$, then U_{TM}, S_{TM} and V_{TM} can be chosen such that they are defined over $\mathbb{R}$. It is sufficient to choose the slices $\tilde{U}_{\mathrm{TM}}(i)$, $\tilde{S}_{\mathrm{TM}}(i)$ and $\tilde{V}_{\mathrm{TM}}(i)$ such that $\tilde{U}_{\mathrm{TM}}(i) = \overline{\tilde{U}}_{\mathrm{TM}}(2-i)$, $\tilde{S}_{\mathrm{TM}}(i) = \overline{\tilde{S}}_{\mathrm{TM}}(2-i)$ and $\tilde{V}_{\mathrm{TM}}(i) = \overline{\tilde{V}}_{\mathrm{TM}}(2-i)$. When the t-scalar dimensions are given by $N = 1, I_1 = 1$, TSVD reduces to the canonical SVD of a matrix in $\mathbb{C}^{D_1 \times D_2}$. The properties of the SVD can be used to show that the t-matrix S_{TM} in Algorithm 1 is unique. The t-matrices U_{TM} and V_{TM} are not unique.

4.1.2 TSVD Approximation

TSVD can be used to approximate data. Given a t-matrix $X_{\mathrm{TM}} \in C^{D_1 \times D_2}$, let $Q \doteq \min(D_1, D_2)$ and let the TSVD of X_{TM} be computed as in Eq. (26). The low-rank approximation $\hat{X}_{\mathrm{TM}}$ of X_{TM} with rank of r $(1 \leq r \leq Q)$ is defined by

$$\hat{X}_{\mathrm{TM}} = U_{\mathrm{TM}} \circ \hat{S}_{\mathrm{TM}} \circ V_{\mathrm{TM}}^{\mathcal{H}} \ , \tag{27}$$

where $\hat{S}_{\mathrm{TM}} = \mathrm{diag}(\lambda_{T,1}, \ldots, \lambda_{T,r}, \underbrace{Z_T, \ldots, Z_T}_{Q-r})$ and $\lambda_{T,1}, \ldots, \lambda_{T,r} \neq Z_T$.

When the t-scalar dimensions are given by $N = 1, I_1 = 1$, Eq. (27) reduces to the SVD low-rank approximation to a matrix in $\mathbb{C}^{D_1 \times D_2}$.

Furthermore, we contend that the approximation $\hat{X}_{\mathrm{TM}}$ computed as in Eq. (27) is the solution of the following optimization problem:

$$\begin{aligned} X_{\mathrm{TM}}^{\mathrm{approx}} &= \mathrm{argmin}_{Y_{\mathrm{TM}} \in C^{D_1 \times D_2}} \|X_{\mathrm{TM}} - Y_{\mathrm{TM}}\|_F \\ &\text{subject to } \mathrm{rank}(Y_{\mathrm{TM}}) \leq r \cdot E_T, \end{aligned} \tag{28}$$

where $\|\cdot\|_F$ denotes the generalized Frobenius norm of a t-matrix, which is a nonnegative t-scalar, as defined in Eq. (20). The result $X_{\mathrm{TM}}^{\mathrm{approx}}$ generalizes the Eckart–Young–Mirsky theorem [7].

To have an optimization problem in the form of (28), the notation $\mathrm{rank}(\cdot)$, i.e., the rank of a t-matrix, and $\min(\cdot)$, i.e., the minimization of a nonnegative t-scalar variable belonging to a subset of S^{nonneg}, and the ordering relationship $\leq$ between two nonnegative t-scalars need to be defined.

These definitions generalize their canonical counterparts. The definitions and the generalized Eckart–Young–Mirsky theorem are discussed in the Appendix.

4.2 THOSVD: Tensor Higher-Order SVD

In multilinear algebra, the higher-order singular value decomposition (HOSVD), also known as the orthogonal Tucker decomposition of a tensor, is a generalization of the SVD. It is commonly used to extract directional information from

Algorithm 2 THOSVD

Input: $X_{GT} \in C^{D_1 \times D_2 \times \cdots \times D_M}$.
Outputs: $U_{\mathrm{TM},1}, U_{\mathrm{TM},2}, \ldots, U_{\mathrm{TM},M}$ and S_{GT} as in Eq. (29)
1: **for all** $1 \le k \le M$ **do**
2: Construct the generalized mode-k flattening $X_{GT(k)} \in C^{D_k \times D_k^{-1} \prod_{m=1}^{M} D_m}$.
3: $(U_{\mathrm{TM},k}, \sim, \sim) \leftarrow \mathtt{tsvd}(X_{GT(k)})$
4: **end for**
5: $S_{GT} \leftarrow X_{GT} \circ_1 U_{\mathrm{TM},1}^{\mathcal{H}} \circ_2 U_{\mathrm{TM},2}^{\mathcal{H}} \cdots \circ_M U_{\mathrm{TM},M}^{\mathcal{H}}$
6: **return** $U_{\mathrm{TM},1}, U_{\mathrm{TM},2}, \ldots, U_{\mathrm{TM},M}$ and S_{GT}.

multi-way arrays [6,30]. The applications of HOSVD include data analytics [29,32], machine learning [23,33,34], DNA and RNA analysis [25,26] and texture mapping in computer graphics [35].

On using the t-scalar algebra, the HOSVD can be generalized further to obtain a tensorial HOSVD, called THOSVD. The THOSVD is obtained by replacing the complex number elements of each multi-way array by t-scalar elements. Based on the definitions of g-tensors in Sect. 3.5, the THOSVD of $X_{GT} \in C^{D_1 \times D_2 \times \cdots \times D_M}$ is given by the following generalized mode-k multiplications:

$$X_{GT} = S_{GT} \circ_1 U_{\mathrm{TM},1} \circ_2 U_{\mathrm{TM},2} \cdots \circ_M U_{\mathrm{TM},M}, \tag{29}$$

where $S_{GT} \in C^{Q_1 \times Q_2 \times \cdots \times Q_M}$ is called the core g-tensor, $U_{\mathrm{TM},k} \in C^{D_k \times Q_k}$ is the mode-k factor t-matrix and $Q_k \doteq \min(D_k, D_k^{-1} \prod_{m=1}^{M} D_m)$ for $1 \le k \le M$.

Given a g-tensor $X_{GT} \in C^{D_1 \times D_2 \times \cdots \times D_M}$, the THOSVD of X_{GT}, as in Eq. (29), is obtained in Algorithm 2, using a strategy analogous to that of Tucker [30] and De Lathauwer et al. [6] for computing the HOSVD of a tensor with elements in $\mathbb{R}$ or $\mathbb{C}$.

Note that THOSVD generalizes the HOSVD for canonical tensors, TSVD for t-matrices and SVD for canonical matrices. Many SVD- and HOSVD-based algorithms can be generalized by TSVD and THOSVD, respectively.

5 Tensor-Based Algorithms

Three tensor-based algorithms are proposed. They are tensorial principal component analysis (TPCA), tensorial two-dimensional principal component analysis (T2DPCA) and tensorial Grassmannian component analysis (TGCA). TPCA and T2DPCA are generalizations of the well-known algorithms PCA and 2DPCA [37]. TGCA is a generalization of the recent GCA algorithm [13,14]. It is possible to generalize many other linear or multilinear algorithms using similar methods.

5.1 TPCA: Tensorial Principal Component Analysis

Principal component analysis (PCA) is a well-known algorithm for extracting the prominent components of observed vectors. PCA is generalized to TPCA in a straightforward manner. Let $X_{\mathrm{TV},1}, \ldots, X_{\mathrm{TV},K} \in C^D$ be K given t-vectors. Then, the covariance-like t-matrix $G_{\mathrm{TM}} \in C^{D \times D}$ is defined by

$$G_{\mathrm{TM}} = \frac{1}{K-1} \sum_{k=1}^{K} (X_{\mathrm{TV},k} - \bar{X}_{\mathrm{TV}}) \circ (X_{\mathrm{TV},k} - \bar{X}_{\mathrm{TV}})^{\mathcal{H}}, \tag{30}$$

where $\bar{X}_{\mathrm{TV}} = (1/K) \sum_{k=1}^{K} X_{\mathrm{TV},k}$. It is not difficult to verify that G_{TM} is Hermitian, namely $G_{\mathrm{TM}}^{\mathcal{H}} = G_{\mathrm{TM}}$.

The t-matrix $U_{\mathrm{TM}} \in C^{D \times D}$ is computed from the TSVD of G_{TM} as in Algorithm 1. Then, given any t-vector $Y_{\mathrm{TV}} \in C^D$, its feature t-vector $Y_{\mathrm{TV}}^{\mathrm{feat}} \in C^D$ is defined by

$$Y_{\mathrm{TV}}^{\mathrm{feat}} = U_{\mathrm{TM}}^{\mathcal{H}} \circ (Y_{\mathrm{TV}} - \bar{X}_{\mathrm{TV}}) \ . \tag{31}$$

To reduce $Y_{\mathrm{TV}}^{\mathrm{feat}}$ from a t-vector in C^D to a t-vector in C^d $(D > d)$, simply discard the last $(D - d)$ t-scalar entries of $Y_{\mathrm{TV}}^{\mathrm{feat}}$.

In algebraic terminology, the column t-vectors of U_{TM} span a linear sub-module of t-vectors, which is a generalization of a vector subspace [3]. In this sense, each t-scalar entry of $Y_{\mathrm{TV}}^{\mathrm{feat}}$ is a generalized coordinate of the projection of the t-vector $(Y_{\mathrm{TV}} - \bar{X}_{\mathrm{TV}})$ onto the sub-module. The low-rank reconstruction $Y_{\mathrm{TV}}^{\mathrm{rec}} \in C^D$ with the parameter d is given by

$$Y_{\mathrm{TV}}^{\mathrm{rec}} = (U_{\mathrm{TM}})_{:,1:d} \circ (Y_{\mathrm{TV}}^{\mathrm{feat}})_{1:d} + \bar{X}_{\mathrm{TV}}, \tag{32}$$

where $(U_{\mathrm{TM}})_{:,1:d} \in C^{D \times d}$ denotes the t-matrix containing the first d t-vector columns of $U_{\mathrm{TM}} \in C^{D \times D}$ and $(Y_{\mathrm{TV}}^{\mathrm{feat}})_{1:d} \in C^d$ denotes the t-vector containing the first d t-scalar entries of $Y_{\mathrm{TV}}^{\mathrm{feat}} \in C^D$.

Note that PCA is a special case of TPCA. When the t-scalar dimensions are given by $N = 1$, $I_1 = 1$, TPCA reduces to PCA.

5.2 T2DPCA: Tensorial Two-Dimensional Principal Component Analysis

The algorithm 2DPCA is an extension of PCA proposed by Yang et al. [37] for analyzing the principal components of matrices. Although 2DPA is written in a non-centered row-vector-oriented form in the original paper [37], it is rewritten here in a centered column-vector-oriented form, which is consistent with the formulation of PCA. The centered column-vector-oriented form of 2DPCA is chosen for discussing its generalization to T2DPCA (Tensorial 2DPCA).

Similar to TPCA, T2DPCA also finds sub-modules, but they are obtained by analyzing t-matrices. Let $X_{\mathrm{TM},1}, \ldots, X_{\mathrm{TM},K} \in C^{D_1 \times D_2}$ be the K observed t-matrices. Then, the Hermitian covariance-like t-matrix $G_{\mathrm{TM}} \in C^{D_1 \times D_1}$ is given by

$$G_{\mathrm{TM}} = \frac{1}{K-1} \sum_{k=1}^{K} (X_{\mathrm{TM},k} - \bar{X}_{\mathrm{TM}}) \circ (X_{\mathrm{TM},k} - \bar{X}_{\mathrm{TM}})^{\mathcal{H}}, \tag{33}$$

where $\bar{X}_{\mathrm{TM}} = (1/K) \sum_{k=1}^{K} X_{\mathrm{TM},k}$.

Then, the t-matrix $U_{\mathrm{TM}} \in C^{D_1 \times D_1}$ is computed from the TSVD of G_{TM} as in Algorithm 1. Given any t-matrix $Y_{\mathrm{TM}} \in C^{D_1 \times D_2}$, its feature t-matrix $Y_{\mathrm{TM}}^{\mathrm{feat}} \in C^{D_1 \times D_2}$ is a centered t-matrix projection (i.e., a collection of centered column t-vector projections) on the module spanned by U_{TM}, namely

$$Y_{\mathrm{TM}}^{\mathrm{feat}} = U_{\mathrm{TM}}^{\mathcal{H}} \circ (Y_{\mathrm{TM}} - \bar{X}_{\mathrm{TM}}) \ . \tag{34}$$

To reduce $Y_{\mathrm{TM}}^{\mathrm{feat}}$ from a t-matrix in $C^{D_1 \times D_2}$ to a t-matrix in $C^{d \times D_2}$ ($D_1 > d$), simply discard the last $(D_1 - d)$ row t-vectors of $Y_{\mathrm{TM}}^{\mathrm{feat}}$.

The T2DPCA reconstruction with the parameter d is given by $Y_{\mathrm{TM}}^{\mathrm{rec}} \in C^{D_1 \times D_2}$ as follows:

$$Y_{\mathrm{TM}}^{\mathrm{rec}} = U_{\mathrm{TM}:,1:d} \circ (Y_{\mathrm{TM}}^{\mathrm{feat}})_{1:d,:} + \bar{X}_{\mathrm{TM}}, \tag{35}$$

where $(U_{\mathrm{TM}})_{:,1:d} \in C^{D_1 \times d}$ denotes the t-matrix containing the first d column t-vectors of U_{TM} and $(Y_{\mathrm{TM}}^{\mathrm{feat}})_{1:d,:} \in C^{d \times D_2}$ denotes the t-matrix containing the first d row t-vectors of $Y_{\mathrm{TM}}^{\mathrm{feat}}$.

When the t-scalar dimensions are given by $N = 1, I_1 = 1$, T2DPCA reduces to 2DPCA. In addition, TPCA is a special case of T2DPCA. When $D_2 = 1$, T2DPCA reduces to TPCA. Furthermore, when $N = 1, I_1 = 1$ and $D_2 = 1$, T2DPCA reduces to PCA.

5.3 TGCA: Tensorial Grassmannian Component Analysis

A t-matrix algorithm which generalizes the recent algorithm for Grassmannian component analysis (GCA) is proposed. An example of GCA can be found in [13], where it forms part of an algorithm for sparse coding on Grassmannian manifolds. In this section, GCA is extended to its generalized version called TGCA (tensorial GCA).

In TGCA, each measurement is a set of t-vectors organized into a "thin" t-matrix, with the number of rows larger than the number of columns. Let $X_{\mathrm{TM},1}, \ldots, X_{\mathrm{TM},K} \in C^{D \times d}$ ($D > d$) be the observed t-matrices. Then, the t-vector columns of each t-matrix are first orthogonalized. Using the t-scalar algebra, it is straightforward to generalize the classical Gram–Schmidt orthogonalization process for t-vectors. The TSVD can also be used to orthogonalize a set of t-vectors. In GCA and TGCA, the choice of orthogonalization algorithm does not matter as long as the algorithm is consistent for all sets of vectors and t-vectors.

Algorithm 3 Generalized Gram–Schmidt orthogonalization

Input: $Y_{\mathrm{TM}} \in C^{D \times d}$.
Outputs: $\dot{Y}_{\mathrm{TM}} \in C^{D \times d}$ satisfying $\dot{Y}_{\mathrm{TM}}^{\mathcal{H}} \circ \dot{Y}_{\mathrm{TM}} = I_{\mathrm{TM}}^{(d)}$.
1: $(\dot{Y}_{\mathrm{TM}})_{:,1} \leftarrow \|(Y_{\mathrm{TM}})_{:,1}\|_F^{-1} \circ (Y_{\mathrm{TM}})_{:,1}$
2: **for all** $2 \le k \le d$ **do**
3: $\quad (\dot{Y}_{\mathrm{TM}})_{:,k} \leftarrow (Y_{\mathrm{TM}})_{:,k}$
4: $\quad$ **for all** $1 \le j \le k-1$ **do**
5: $\quad\quad (\dot{Y}_{\mathrm{TM}})_{:,k} \leftarrow (\dot{Y}_{\mathrm{TM}})_{:,k} - \langle (\dot{Y}_{\mathrm{TM}})_{:,k}, (\dot{Y}_{\mathrm{TM}})_{:,j} \rangle \circ (\dot{Y}_{\mathrm{TM}})_{:,j}$
6: $\quad$ **end for**
7: $\quad (\dot{Y}_{\mathrm{TM}})_{:,k} \leftarrow \|(\dot{Y}_{\mathrm{TM}})_{:,k}\|_F^{-1} \circ (\dot{Y}_{\mathrm{TM}})_{:,k}$
8: **end for**
9: **return** $\dot{Y}_{\mathrm{TM}}$

Given a t-matrix $Y_{\mathrm{TM}} \in C^{D \times d}$, let $\dot{Y}_{\mathrm{TM}} \in C^{D \times d}$ be the corresponding unitary orthogonalized t-matrix (namely, $\dot{Y}_{\mathrm{TM}}^{\mathcal{H}} \circ \dot{Y}_{\mathrm{TM}} = I_{\mathrm{TM}}^{(d)}$) computed from Y_{TM}. Let $(Y_{\mathrm{TM}})_{:,k}$ be the kth column t-vector of Y_{TM}, and let $(\dot{Y}_{\mathrm{TM}})_{:,k}$ be the kth column t-vector of $\dot{Y}_{\mathrm{TM}}$ for $1 \le k \le d$. The generalized Gram–Schmidt orthogonalization is given by Algorithm 3.

Let $\dot{X}_{\mathrm{TM},k} \in C^{D \times d}$ be the unitary orthogonalized t-matrices computed from $X_{\mathrm{TM},k}$ for $1 \le k \le K$. Then, for $1 \le k, k' \le K$, the (k, k') t-scalar entry of the symmetric t-matrix $G_{\mathrm{TM}} \in C^{K \times K}$ is nonnegative and given by

$$(G_{\mathrm{TM}})_{k,k'} = \|\dot{X}_{\mathrm{TM},k}^{\mathcal{H}} \circ \dot{X}_{\mathrm{TM},k'}\|_F^2 \ , \quad 1 \le k, k' \le K, \tag{36}$$

where $\|\cdot\|_F$ is the generalized Frobenius norm of a t-matrix, as defined by Eq. (20).

Given any query t-matrix sample $Y_{\mathrm{TM}} \in C^{D \times d}$, let $\dot{Y}_{\mathrm{TM}} \in C^{D \times d}$ be the corresponding unitary orthogonalized t-matrix computed from Y_{TM}. Then, the kth t-scalar entry of $K_{\mathrm{TV}} \in C^K$ is computed as follows:

$$(K_{\mathrm{TV}})_k = \|\dot{Y}_{\mathrm{TM}}^{\mathcal{H}} \circ \dot{X}_{\mathrm{TM},k}\|_F^2, \quad 1 \le k \le K. \tag{37}$$

Since G_{TM}, computed as in Eq. (36), is symmetric, the TSVD of G_{TM} has the following form:

$$G_{\mathrm{TM}} = U_{\mathrm{TM}} \circ S_{\mathrm{TM}} \circ U_{\mathrm{TM}}^{\mathcal{H}} \ . \tag{38}$$

Furthermore, if it is assumed that the diagonal entries $S_{\mathrm{TM}} \doteq \mathrm{diag}(\lambda_{T,1}, \cdots, \lambda_{T,K})$ are all strictly positive, then the multiplicative inverse of $\lambda_{T,k}$ exists for $1 \le k \le K$. The t-matrix $S_{\mathrm{TM}}^{1/2} \doteq \mathrm{diag}(\sqrt{\lambda_{T,1}}, \cdots, \sqrt{\lambda_{T,K}})$ is called the t-matrix square root of S_{TM}, and the t-matrix $S_{\mathrm{TM}}^{-1/2} \doteq \mathrm{diag}(\frac{E_T}{\sqrt{\lambda_{T,1}}}, \cdots, \frac{E_T}{\sqrt{\lambda_{T,K}}})$ is called the inverse t-matrix of $S_{\mathrm{TM}}^{1/2}$.

Thus, the features of the t-matrix sample $Y_{\mathrm{TM}} \in C^{D\times d}$ are given by the t-vector $Y_{\mathrm{TV}}^{\mathrm{feat}} \in C^K$ as

$$Y_{\mathrm{TV}}^{\mathrm{feat}} = S_{\mathrm{TM}}^{-1/2} \circ U_{\mathrm{TM}}^{\mathcal{H}} \circ K_{\mathrm{TV}} \tag{39}$$

and the features of the kth measurement $X_{\mathrm{TM},k}$ are given by the t-vector $X_{\mathrm{TV},k}^{\mathrm{feat}}$ as follows:

$$X_{\mathrm{TV},k}^{\mathrm{feat}} = S_{\mathrm{TM}}^{-1/2} \circ U_{\mathrm{TM}}^{\mathcal{H}} \circ (G_{\mathrm{TM}})_{:,k} \ \ , 1 \le k \le K, \tag{40}$$

where $(G_{\mathrm{TM}})_{:,k}$ denotes the kth t-vector column of G_{TM}. It is not difficult to verify that $S_{\mathrm{TM}}^{-1/2} \circ U_{\mathrm{TM}}^{\mathcal{H}} \circ G_{\mathrm{TM}} \equiv S_{\mathrm{TM}}^{1/2} \circ U_{\mathrm{TM}}^{\mathcal{H}}$. This yields the following compact form for $X_{\mathrm{TV},k}^{\mathrm{feat}}$.

$$X_{\mathrm{TV},k}^{\mathrm{feat}} = (S_{\mathrm{TM}}^{1/2} \circ U_{\mathrm{TM}}^{\mathcal{H}})_{:,k} \ \ , 1 \le k \le K, \tag{41}$$

where $(S_{\mathrm{TM}}^{1/2} \circ U_{\mathrm{TM}}^{\mathcal{H}})_{:,k}$ denotes the kth t-vector column of the t-matrix $(S_{\mathrm{TM}}^{1/2} \circ U_{\mathrm{TM}}^{\mathcal{H}})$. Equation (41) is more efficient in computations than Eq. (40).

The dimension of a TGCA feature t-vector is reduced from K to K' ($K > K'$) by discarding the last $(K - K')$ t-scalar entries. It is noted that GCA is a special case of TGCA when the dimensions of the t-scalars are given by $N = 1$, $I_1 = 1$.

6 Experiments

The results obtained from TSVD, THOSVD, TPCA, T2DPCA, TGCA and their precursors are compared in applications to low-rank approximation in Sect. 6.1, reconstruction in Sect. 6.2 and supervised classification of images in Sect. 6.3.

In these experiments, "vertical" and "horizontal" comparisons between generalized algorithms and the corresponding canonical algorithms are made.

In a "vertical" experiment, tensorized data are obtained from the canonical data in 3×3 neighborhoods. The associated t-scalar is a 3×3 array. To make the vertical comparison fair, the central slices of a generalized result are put into the original canonical form and then compared with the result of the associated canonical algorithm.

In a "horizontal" comparison, a generalized order-N array of order-two t-scalars is equivalent to a canonical order-$(N+2)$ array of scalars. Therefore, a generalized algorithm based on order-N arrays of order-two t-scalars is compared with a canonical algorithm based on order-$(N+2)$ arrays of scalars.

6.1 Low-Rank Approximation

TSVD approximation is computed as in Eq. (27). THOSVD approximation generalizes low-rank approximation by TSVD and low-rank approximation by HOSVD. To simplify the calculations, the approximation is obtained for a g-tensor X_{GT} in $C^{D_1\times D_2\times D_3}$. Let $Q_k \doteq \min(D_k, D_k^{-1} D_1 D_2 D_3)$ for $k = 1, 2, 3$. The THOSVD of X_{GT} yields

$$X_{GT} = S_{GT} \circ_1 U_{\mathrm{TM},1} \circ_2 U_{\mathrm{TM},2} \circ_3 U_{\mathrm{TM},3}, \tag{42}$$

where $U_{\mathrm{TM},k} \in C^{D_k\times Q_k}$ for $k = 1, 2, 3$ and $S_{\mathrm{TM}} \in C^{Q_1\times Q_2\times Q_3}$.

The low-rank approximation $\hat{X}_{GT} \in C^{D_1\times D_2\times D_3}$ to X_{GT} and with multilinear rank tuple (r_1, r_2, r_3), ($1 \le r_k \le Q_k$ for all $k = 1, 2, 3$), is computed as in Eq. (43), where $(U_{\mathrm{TM},k})_{:,1:r_k}$ denotes the t-matrix containing the first r_k t-vector columns of $U_{\mathrm{TM},k}$ for $k = 1, 2, 3$ and $(S_{GT})_{1:r_1,1:r_2,1:r_3} \in C^{r_1\times r_2\times r_3}$ denotes the g-tensor containing the first $r_1 \times r_2 \times r_3$ t-scalar entries of S_{GT}.

$$\begin{aligned}\hat{X}_{GT} = (S_{GT})_{1:r_1,1:r_2,1:r_3} &\circ_1 (U_{\mathrm{TM},1})_{:,1:r_1} \\ &\circ_2 (U_{\mathrm{TM},2})_{:,1:r_2} \circ_3 (U_{\mathrm{TM},3})_{:,1:r_3} \ .\end{aligned} \tag{43}$$

When the t-scalar dimensions are given by $N = 1$, $I_1 = 1$, Eq. (43) reduces to the HOSVD low-rank approximation of a tensor in $\mathbb{C}^{D_1\times D_2\times D_3}$. When the g-tensor dimension $D_3 = 1$, Eq. (43) reduces to the SVD low-rank approximation of a canonical matrix in $\mathbb{C}^{D_1\times D_2}$.

6.1.1 TSVD Versus SVD—A "Vertical" Comparison

The low-rank approximation performances of TSVD and SVD are compared. In the experiment, the test sample is the $512 \times 512 \times 3$ RBG Lena image downloaded from Wikipedia.[1]

For the SVD low-rank approximations, the RGB Lena image is split into three 512×512 monochrome images. Each monochrome image is analyzed using the SVD. The three extracted monochrome Lena images are order-two arrays in $\mathbb{R}^{512\times512}$. Each monochrome Lena image is tensorized to produce a t-image (a generalized monochrome image) in $R^{512\times512} \equiv \mathbb{R}^{3\times3\times512\times512}$. In the tensorized version of the image, each pixel value is replaced by a 3×3 square of values obtained from the 3×3 neighborhood of the pixel. Padding with 0 is used where necessary at the boundary of the image.

To evaluate the TSVD approximations in a manner relevant to the SVD approximations, upon obtaining a t-image approximation $\hat{X}_{\mathrm{TM}} \in \mathbb{R}^{3\times3\times512\times512}$, the part $\hat{X}_{MT}(i)|_{i=(2,2)} \in \mathbb{R}^{512\times512}$, i.e., the central slice of the TSVD approximation, is used for comparisons.

Given an array X of any order over the real numbers $\mathbb{R}$, let $\hat{X}$ be an approximation to X. Then, the PSNR (peak signal-to-noise ratio) for $\hat{X}$ is defined as in [1] by

$$\mathrm{PSNR} = 20 \log_{10} \frac{\mathrm{MAX} \cdot \sqrt{N^{\mathrm{entry}}}}{\|X - \hat{X}\|_F}, \tag{44}$$

[1] https://en.wikipedia.org/wiki/Lenna.

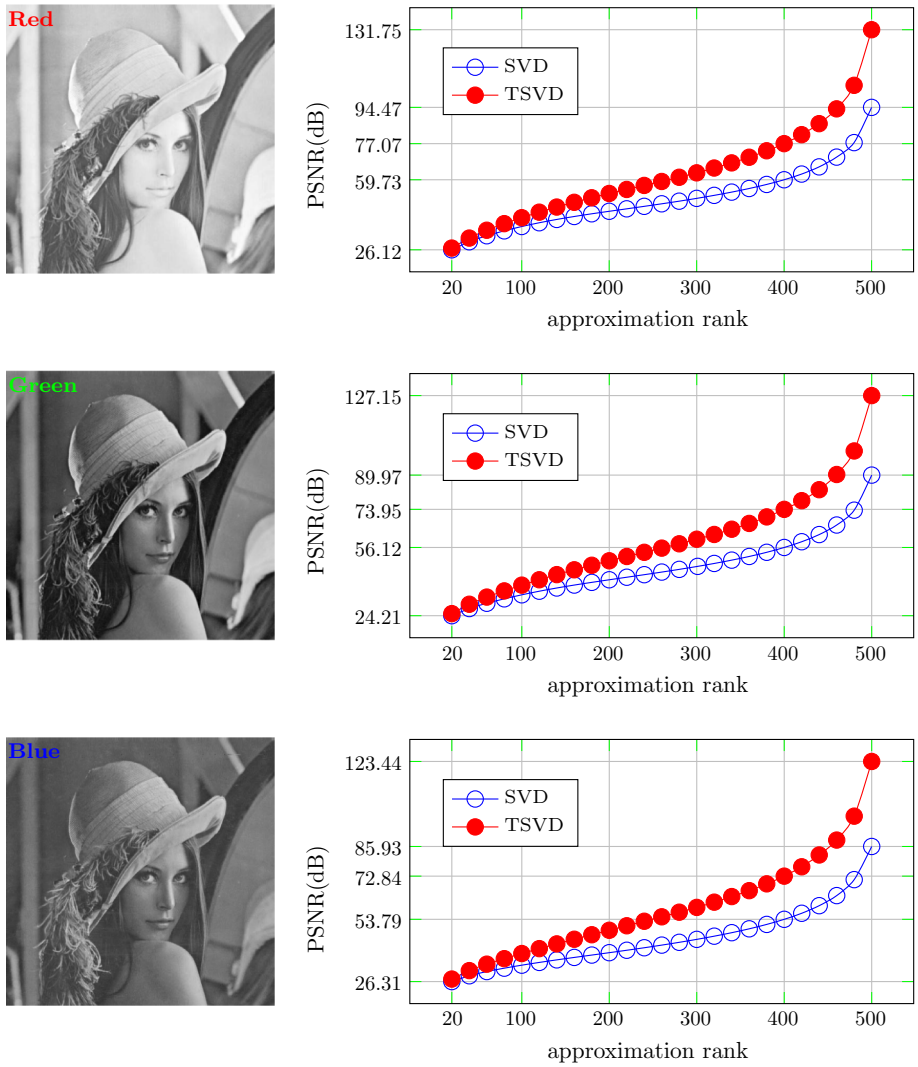


Fig. 1 A "vertical" comparison of low-rank approximations by SVD and TSVD for each monochrome Lena image. First column: monochrome images extracted from the RGB Lena image. Second column: PSNR curves of SVD/TSVD approximation on/for each monochrome image

where N^{entry} denotes the number of real number entries of X, $\|X - \hat{X}\|_F$ is the canonical Frobenius norm of the array $(X - \hat{X})$ and MAX is the maximum possible value of the entries of X. In all the experiments, MAX = 255. In this experiment comparing TSVD and SVD, $N^{\text{entry}} = 512 \times 512 = 262144$.

Figure 1 shows the PSNR curves of the SVD and TSVD approximations as functions of the rank of $\hat{X}$. It is clear that the PSNR of the TSVD approximation is consistently higher than that of SVD approximation. When the rank $r = 500$, the PSNRs of TSVD and SVD differ by more than 37 dBs.

6.1.2 TSVD Versus HOSVD—A "Horizontal" Comparison

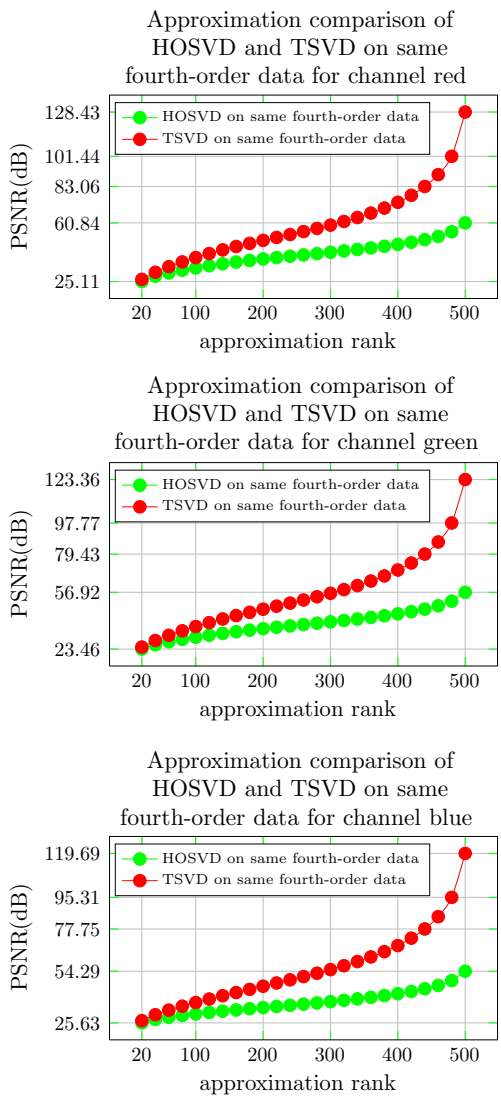
Given a monochrome Lena image as an order-two array in $\mathbb{R}^{512\times512}$ and its tensorized form as an order-four array in $\mathbb{R}^{3\times3\times512\times512}$, TSVD yields an approximation array in $\mathbb{R}^{3\times3\times512\times512}$. Since the HOSVD is applicable to order-four arrays in $\mathbb{R}^{3\times3\times512\times512}$, we give a "horizontal" comparison of the performances of TSVD and HOSVD.

More specifically, given a generalized monochrome Lena image $X_{\text{TM}} \equiv X \in C^{512\times512} \equiv \mathbb{R}^{3\times3\times512\times512}$ and a specified rank r, the TSVD approximation yields a t-matrix $\hat{X}_{\text{TM}} \in C^{512\times512} \equiv \mathbb{R}^{3\times3\times512\times512}$, which is computed as in Eq. (27) with $D_1 = 512$ and $D_2 = 512$.

Let the HOSVD of $X \in \mathbb{R}^{3\times3\times512\times512}$ be $X = S \times_1 U_1 \times_2 U_2 \times_3 U_3 \times_4 U_4$ where $S \in \mathbb{R}^{3\times3\times512\times512}$ denotes the core tensor, and $U_1 \in \mathbb{R}^{3\times3}$, $U_2 \in \mathbb{R}^{3\times3}$, $U_3 \in \mathbb{R}^{512\times512}$, $U_4 \in \mathbb{R}^{512\times512}$ are all orthogonal matrices. Then, to give a "horizontal" comparison with the TSVD approximation $\hat{X}_{\text{TM}}$ with rank r, the HOSVD approximation $\hat{X} \in \mathbb{R}^{3\times3\times512\times512}$ is given by the multi-mode product

$$\hat{X} = (S)_{:,:,1:r,1:r} \times_1 U_1 \times_2 U_2 \times_3 (U_3)_{:,1:r} \times_4 (U_4)_{:,1:r} \ . \tag{45}$$

The PSNRs TSVD and HOSVD are computed as in Eq. (44) with MAX = 255 and $N^{\text{entry}} = 3 \times 3 \times 512 \times 512 = 2359296$.



PSNRs of HOSVD/TSVD approximations (on same fourth-order data) with different approximation rank r

channel type	rank r	PSNR (dB) HOSVD	PSNR (dB) TSVD
red	20	25.11	26.44
	60	30.37	34.18
	100	33.42	39.73
	140	35.92	44.42
	180	37.96	48.37
	220	39.72	51.98
	260	41.33	55.57
	300	42.94	59.51
	340	44.65	64.14
	380	46.62	69.89
	420	49.08	77.68
	460	52.62	90.29
	500	60.84	128.43
green	20	23.46	24.62
	60	27.76	31.55
	100	30.44	36.71
	140	32.69	41.19
	180	34.61	45.11
	220	36.31	48.75
	260	37.90	52.38
	300	39.48	56.31
	340	41.17	60.88
	380	43.09	66.56
	420	45.49	74.18
	460	48.94	86.58
	500	56.92	123.36
blue	20	25.63	26.88
	60	28.77	32.71
	100	30.61	36.93
	140	32.13	40.69
	180	33.48	44.23
	220	34.74	47.74
	260	36.03	51.32
	300	37.41	55.18
	340	38.96	59.66
	380	40.77	65.16
	420	43.08	72.64
	460	46.43	84.56
	500	54.29	119.69

Fig. 2 A "horizontal" comparison of low-rank approximations by HOSVD and TSVD on each generalized monochrome Lena image, as a fourth-order real number array in $\mathbb{R}^{3\times3\times512\times512}$. First column: PSNR curves, over rank r, of HOSVD/TSVD approximations on each generalized monochrome Lena image. Second column: some quantitative PSNRs of HOSVD/TSVD approximations with rank r

For each of the generalized monochrome Lena images (respectively marked by the channel type "red," "green" and "blue"), as a $3 \times 3 \times 512 \times 512$ real number array, the PSNRs of TSVD and HOSVD are shown in Fig. 2.

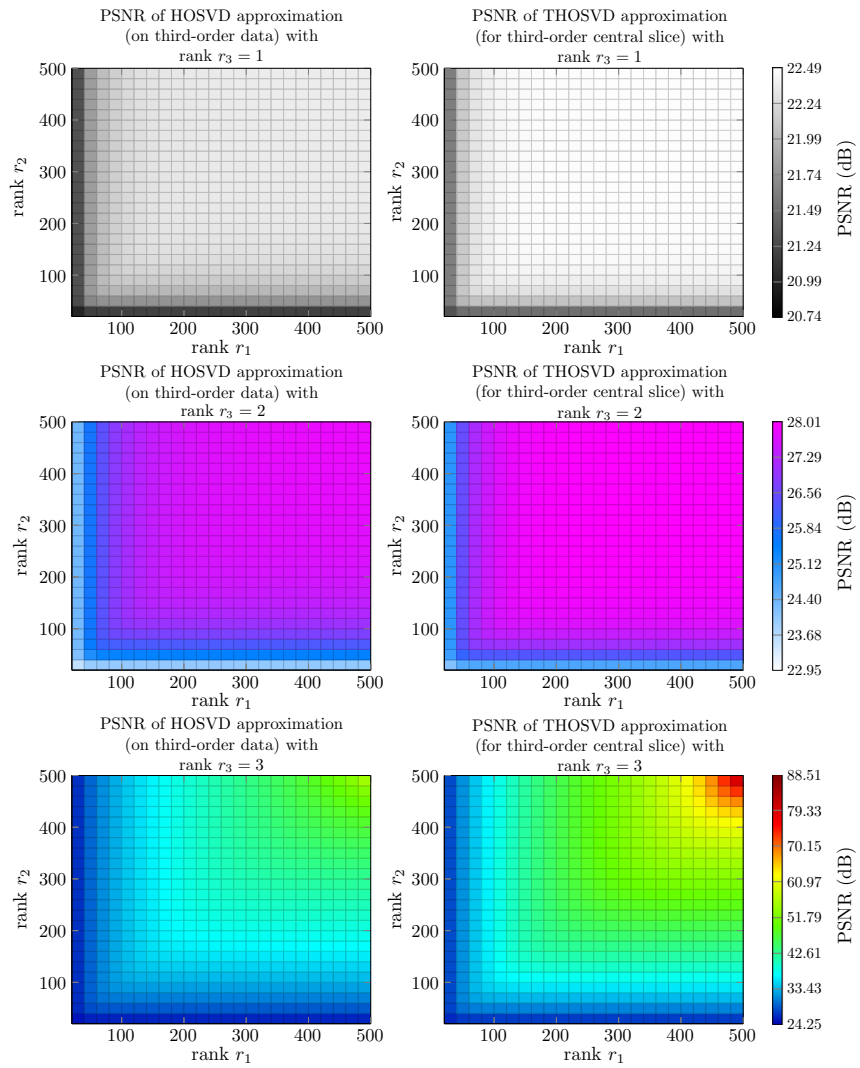
As rank r is varied, the PSNR of TSVD approximation is always higher than that of the corresponding HOSVD approximation. When rank r is equal to 500, the PSNRs of TSVD and HOSVD approximations differ significantly.

6.1.3 THOSVD Versus HOSVD—A "Vertical" Comparison

The low-rank approximation performances of THOSVD and HOSVD are compared. For the HOSVD approximations, the RGB Lena image, which is a tensor in $\mathbb{R}^{512\times512\times3}$, is used as the test sample. For the THOSVD, the 3×3 neighborhood (with zero-padding) strategy is used to tensorize each real number entry of the RGB Lena image. The obtained t-image X_{GT} is a g-tensor in $R^{512\times512\times3}$, i.e., an order-five array in $\mathbb{R}^{3\times3\times512\times512\times3}$.

To give a "vertical" comparison, on obtaining an approximation $\hat{X}_{GT} \in \mathbb{R}^{3\times3\times512\times512\times3}$, $\hat{X}_{GT}(i)|_{i=(2,2)} \in \mathbb{R}^{512\times512\times3}$, i.e., the central slice of the THOSVD approximation, is compared with the HOSVD approximation on the RGB Lena image.

Figure 3 shows a "vertical" comparison of the PSNR maps of THOSVD and HOSVD approximations and the tabulated PSNRs for some representative multilinear rank tuples (r_1, r_2, r_3). It shows that the PSNR of the THOSVD approximation is consistently higher than the PSNR of the HOSVD approximation. When $(r_1, r_2, r_3) = (500, 500, 3)$, the approximations obtained by THOSVD and HOSVD differ by 30.29 dB in their PSNR values.



PSNRs of HOSVD/THOSVD approximations (on/for third-order data/slice) with different multilinear rank tuple (r_1, r_2, r_3)

Multilinear Rank			PSNR (dB)	
r_1	r_2	r_3	HOSVD	THOSVD
20	20	1	20.74	21.05
60	60	1	21.91	22.23
100	100	1	22.19	22.42
140	140	1	22.28	22.47
180	180	1	22.33	22.48
220	220	1	22.35	22.49
260	260	1	22.36	22.49
300	300	1	22.36	22.49
340	340	1	22.37	22.49
380	380	1	22.37	22.49
420	420	1	22.38	22.49
460	460	1	22.38	22.49
500	500	1	22.38	22.49
20	20	2	22.95	23.49
60	60	2	25.82	26.67
100	100	2	26.81	27.48
140	140	2	27.24	27.78
180	180	2	27.45	27.90
220	220	2	27.57	27.95
260	260	2	27.64	27.98
300	300	2	27.70	28.00
340	340	2	27.74	28.00
380	380	2	27.77	28.01
420	420	2	27.79	28.01
460	460	2	27.81	28.01
500	500	2	27.83	28.01
20	20	3	24.25	25.01
60	60	3	29.46	31.78
100	100	3	32.51	35.93
140	140	3	34.75	39.22
180	180	3	36.52	42.03
220	220	3	37.98	44.73
260	260	3	39.37	47.40
300	300	3	40.79	50.03
340	340	3	42.34	52.86
380	380	3	44.14	56.17
420	420	3	46.44	60.68
460	460	3	49.82	68.09
500	500	3	58.22	88.51

Fig. 3 A "vertical" comparison of THOSVD approximations and HOSVD approximations with the multilinear rank tuple (r_1, r_2, r_3). First column: PSNR maps of HOSVD approximation on the RGB Lena image. Second column: PSNR maps of THOSVD approximation for the RGB Lena image (i.e., third-order central slice of THOSVD approximation). Third column: some quantitative PSNRs of HOSVD/THOSVD approximations with representative multilinear rank tuples

6.1.4 THOSVD Versus HOSVD—A "Horizontal Comparison"

Given a fifth-order array $X \in \mathbb{R}^{3\times3\times512\times512\times5}$ tensorized from the RGB Lena image, which is a third-order array in $\mathbb{R}^{512\times512\times3}$, both THOSVD and HOSVD can be applied to the same data X.

THOSVD takes X as a g-tensor $X_{GT} \in C^{512\times512\times3} \equiv \mathbb{R}^{3\times3\times512\times512\times3}$, while HOSVD takes X merely as a canonical fifth-order array in $\mathbb{R}^{3\times3\times512\times512\times3}$.

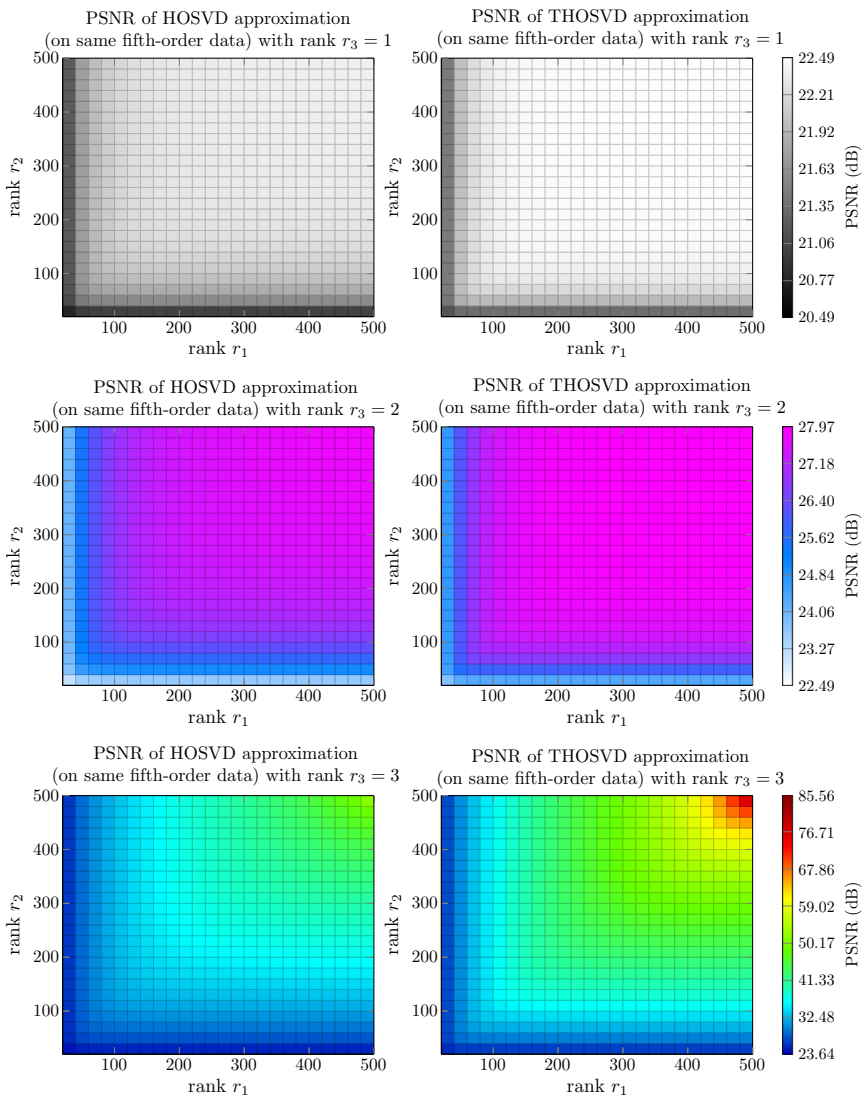
Then, given a rank tuple (r_1, r_2, r_3) subject to $1 \le r_1 \le 512$, $1 \le r_2 \le 512$ and $1 \le r_3 \le 3$, the THOSVD approximation $\hat{X}_{GT} \in C^{512\times512\times3}$ is computed as in Eq. (43).

Let the HOSVD of $X \in \mathbb{R}^{3\times3\times512\times512\times3}$ be $X = S \times_1 U_1 \times_2 U_2 \times_3 U_3 \times_4 U_4 \times_5 U_5$ where $S \in \mathbb{R}^{3\times3\times512\times512\times3}$ is the core tensor and $U_1 \in \mathbb{R}^{3\times3}$, $U_2 \in \mathbb{R}^{3\times3}$, $U_3 \in \mathbb{R}^{512\times512}$, $U_4 \in \mathbb{R}^{512\times512}$, $U_5 \in \mathbb{R}^{3\times3}$ are all orthogonal matrices.

Then, to give a "horizontal" comparison with the THOSVD approximation $\hat{X}_{GT} \in C^{512\times512\times3}$ with a rank tuple (r_1, r_2, r_3), the HOSVD approximation $\hat{X} \in \mathbb{R}^{3\times3\times512\times512\times3}$ is given by the following multi-mode product:

$$\hat{X} = (S)_{:,:,1:r_1,1:r_2,1:r_3} \times_1 U_1 \times_2 U_2 \times_3 (U_3)_{:,1:r_1} \times_4 (U_4)_{:,1:r_2} \times_5 (U_5)_{:,1:r_3} . \tag{46}$$

Figure 4 shows the "horizontal" comparison of THOSV approximations and HOSVD approximations on the same array with different rank tuples (r_1, r_2, r_3). Albeit somewhat



PSNRs of HOSVD/THOSVD approximations (on same fifth-order data) with different multilinear rank tuple (r_1, r_2, r_3)

Multilinear Rank			PSNR (dB)	
r_1	r_2	r_3	HOSVD	THOSVD
20	20	1	20.49	20.88
60	60	1	21.68	22.11
100	100	1	22.01	22.36
140	140	1	22.17	22.44
180	180	1	22.26	22.46
220	220	1	22.30	22.48
260	260	1	22.33	22.48
300	300	1	22.35	22.49
340	340	1	22.36	22.49
380	380	1	22.37	22.49
420	420	1	22.38	22.49
460	460	1	22.39	22.49
500	500	1	22.39	22.49
20	20	2	22.49	23.19
60	60	2	25.19	26.30
100	100	2	26.21	27.24
140	140	2	26.79	27.60
180	180	2	27.13	27.77
220	220	2	27.34	27.85
260	260	2	27.48	27.90
300	300	2	27.57	27.93
340	340	2	27.65	27.95
380	380	2	27.71	27.96
420	420	2	27.76	27.96
460	460	2	27.80	27.97
500	500	2	27.83	27.97
20	20	3	23.64	24.59
60	60	3	28.11	30.71
100	100	3	30.61	34.51
140	140	3	32.62	37.48
180	180	3	34.29	39.93
220	220	3	35.75	42.20
260	260	3	37.09	44.48
300	300	3	38.40	46.92
340	340	3	39.81	49.71
380	380	3	41.44	53.11
420	420	3	43.50	57.69
460	460	3	46.55	65.10
500	500	3	53.96	85.56

Fig. 4 A "horizontal" comparison of THOSVD approximations and HOSVD approximations with multilinear rank tuple (r_1, r_2, r_3). First column: PSNR maps of HOSVD approximation. Second column: PSNR maps of THOSVD approximation. Third column: some PSNRs of HOSVD/THOSVD approximations on the same fifth-order data with representative multilinear rank tuples (r_1, r_2, r_3)

smaller in PSNRs, the results in Fig. 4 are similar to the results in Fig. 3 (a "vertical" comparison), corroborating the claim that a THOSVD approximation outperforms, in terms of PSNR, the corresponding HOSVD approximation on the same data.

6.2 Reconstruction

The qualities of the low-rank reconstructions produced by TPCA and PCA and by T2DPCA and 2DPCA, as described by Eqs. (32) and (35), are compared.

The effectiveness of PCA, 2DPCA, TPCA and T2DPCA for reconstruction is assessed using the ORL dataset. The dataset contains 400 face images in 40 classes, i.e., 10 images/class × 40 classes. Each image has 112 × 92 pixels.[2] The first 200 images (5 images/class × 40 classes) are used as the observed images, and the remaining 200 images are the query images.

For the experiments with TPCA/T2DPCA, all ORL images are tensorized to t-images in $R^{112\times 92}$, namely order-four arrays in $\mathbb{R}^{3\times 3\times 112\times 92}$. Eigendecompositions and t-eigendecompositions are computed on the observed images and t-images, respectively. Reconstructions are computed for the query images and t-images, respectively. The number of PSNRs for the reconstructed images and t-images is 200. It

[2] https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

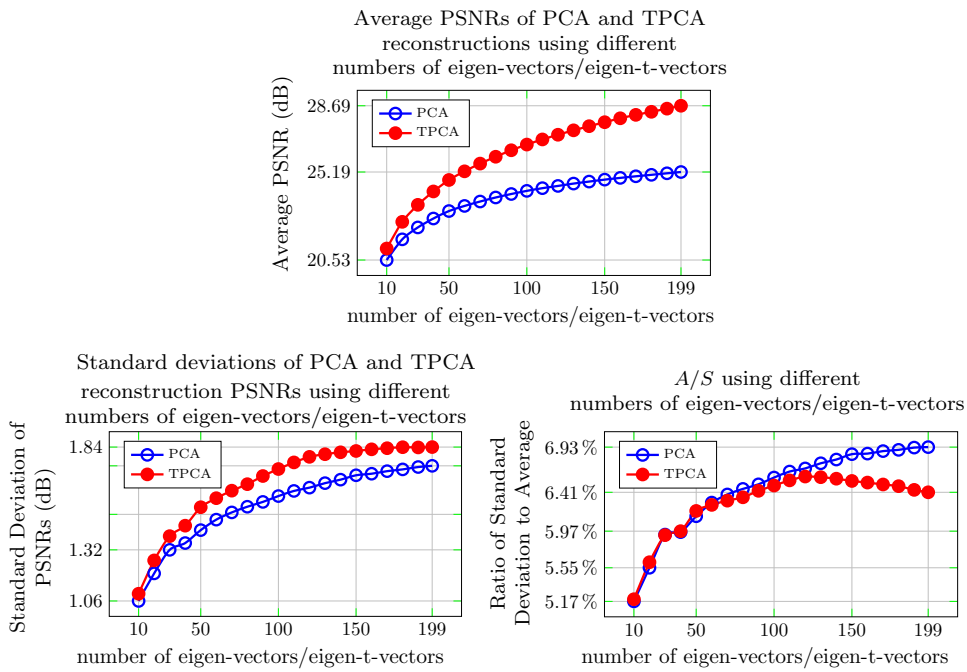


Fig. 5 A "vertical" comparison of PSNR averages and standard deviations for PCA and TPCA reconstructions

Averages and standard deviations
of PSNRs of PCA/TPCA reconstruction on the ORL dataset
d = the number of eigen-vectors/eigen-t-vectors A = Average (dB)
S = Standard deviation (dB).

d		20	40	60	80	100	120	140	160	180	199
A	PCA	21.63	22.73	23.40	23.84	24.19	24.45	24.68	24.88	25.04	25.19
	TPCA	22.55	24.16	25.23	26.00	26.64	27.16	27.61	28.03	28.37	28.69
S	PCA	1.20	1.35	1.47	1.54	1.59	1.64	1.68	1.71	1.73	1.75
	TPCA	1.27	1.44	1.58	1.65	1.73	1.79	1.81	1.83	1.84	1.84
A/S	PCA	5.55	5.96	6.30	6.45	6.58	6.69	6.79	6.86	6.90	6.93
	TPCA	5.62	5.97	6.27	6.36	6.49	6.59	6.57	6.53	6.48	6.41

is convenient to use the average of the PSNRs (denoted by A), the standard deviation of PSNRs (denoted by S) and the ratio, A/S. A larger value of A with a smaller value of S indicates a better quality of reconstruction.

6.2.1 TPCA Versus PCA—A "Vertical" Comparison

To make the TPCA and PCA reconstructions computationally tractable, each image is resized to 56×46 pixels by bi-cubic interpolation. The resized images are also tensorized to t-images, i.e., order-four arrays in $\mathbb{R}^{3\times3\times56\times46}$. The obtained images and t-images are then transformed to vectors and t-vectors, respectively, by stacking their columns. The central slices of the TPCA reconstructions are compared with the PCA reconstructions.

Figure 5 shows graphs and some tabulated values of A, S and A/S for a number of eigenvectors and eigen-t-vectors. Note that K linearly independent observed vectors or t-vectors yield at most $(K-1)$ eigenvectors or eigen-t-vectors. Thus, the maximum number of eigenvectors and eigen-t-vectors in Fig. 5 is 199 ($K = 200$).

The average PSNR for TPCA is consistently higher than the average PSNR for PCA. The PSNR standard deviation for TPCA is slightly larger than the PSNR standard deviation for PCA, but the ratio A/S for TPCA is generally smaller than the ratio A/S for PCA. This indicates that TPCA outperforms PCA in terms of reconstruction quality.

6.2.2 T2DPCA Versus 2DPCA—A "Vertical" Comparison

The same observed samples from the ORL dataset (the first 200 images, 5 images/class $\times$ 40 classes) and query samples (the remaining 200 images) are used to compare the reconstruction performances of T2DPCA and 2DPCA. The central slices of the T2DPCA are compared with the 2DPCA reconstructions.

Figure 6 shows the reconstruction curves and some tabulated values yielded by T2PCA and 2DPCA as functions of the number d of eigenvectors or eigen-t-vectors. The average PSNR obtained by T2DPCA is consistently higher than the average PSNR obtained by 2DPCA. When the parameter d equals 111, the gap between the two average PSNRs is 31.98

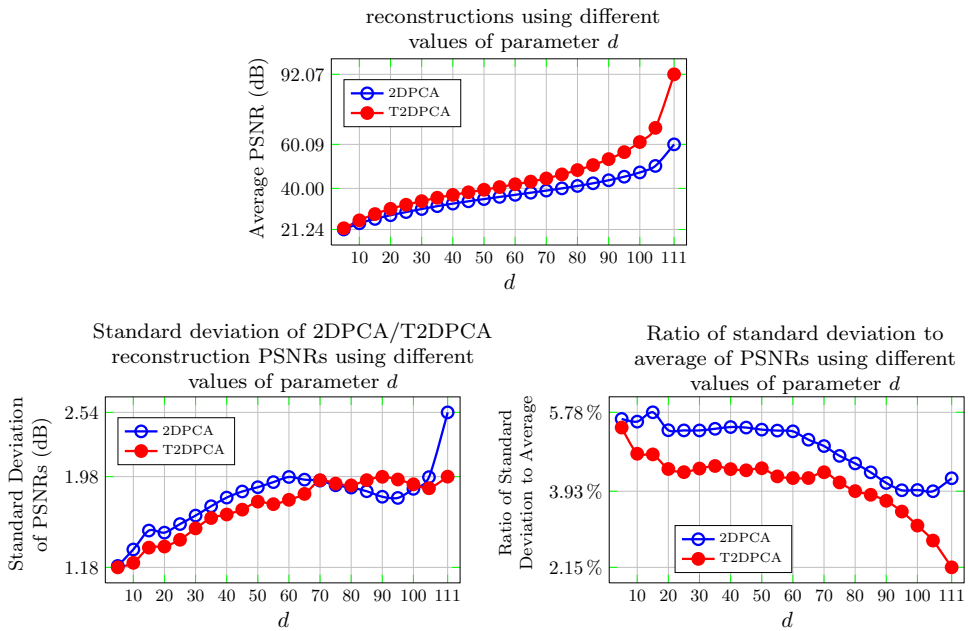


Fig. 6 A "vertical" comparison of PSNR averages and standard deviations for the 2DPCA and T2DPCA reconstructions

Average and standard deviation of PSNRs of 2DPCA/T2DPCA reconstruction on the ORL dataset
d = the number of eigen-vectors/eigen-t-vectors A = Average (dB)
S = Standard deviation (dB) R = Ratio of S to A (%)

	d	5	10	20	30	40	50	60	70	80	90	100	111
A	2DPCA	21.24	24.08	27.75	30.58	33.04	35.11	37.06	38.99	41.11	43.70	47.28	60.09
	T2DPCA	21.80	25.40	30.66	34.17	37.02	39.38	41.89	44.53	48.36	53.36	61.15	92.07
S	2DPCA	1.19	1.34	1.49	1.64	1.80	1.89	1.98	1.94	1.88	1.80	1.87	2.54
	T2DPCA	1.18	1.22	1.36	1.52	1.65	1.76	1.78	1.95	1.90	1.98	1.91	1.98
A/S	2DPCA	5.62	5.56	5.36	5.35	5.43	5.37	5.33	4.99	4.58	4.13	3.96	4.24
	T2DPCA	5.42	4.81	4.45	4.46	4.45	4.47	4.24	4.38	3.93	3.71	3.13	2.15

dBs. Furthermore, the PSNR standard deviation for T2DPCA is also generally smaller than the PSNR standard deviation for 2DPCA. In terms of reconstruction quality, T2DPCA outperforms 2DPCA.

6.3 Classification

TGCA and GCA are applied to the classification of the pixel values in hyperspectral images. Hyperspectral images have hundreds of spectral bands, in contrast with RGB images which have only three spectral bands. The multiple spectral bands and high resolution make hyperspectral imagery essential in remote sensing, target analysis, classification and identification [10,15,21,24,36,38,40]. Two publicly available datasets are used to evaluate the effectiveness of TGCA and GCA for supervised classification.

6.3.1 Datasets

The first hyperspectral image dataset is the Indian Pines cube (Indian cube for short), which consists of 145 × 145 hyperspectral pixels (hyperpixels for short) and has 220 spectral bands, yielding an array of order three in $\mathbb{R}^{145\times145\times220}$. The Indian cube comes with ground-truth labels for 16 classes [31]. The second hyperspectral image dataset is the Pavia University cube (Pavia cube for short), which consists of 610 × 340 hyperpixels with 103 spectral bands, yielding an array of order three in $\mathbb{R}^{610\times340\times103}$. The ground truth contains 9 classes [31].

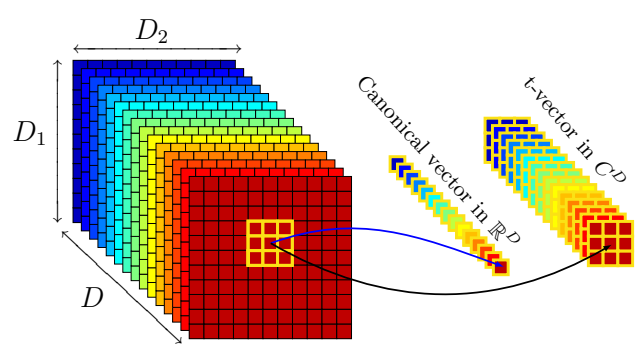


Fig. 7 Tensorization of a canonical vector extracted from a hyperspectral cube

6.3.2 Tensorization

Given a hyperspectral cube, let D_1 be the number of rows, D_2 the number of columns and D the number of spectral bands. A hyperpixel is represented by a vector in $\mathbb{R}^D$. Each pixel is tensorized by its 3×3 neighborhood. The tensorized hyperspectral cube is represented by an array in $\mathbb{R}^{3\times 3\times D_1\times D_2\times D}$. Each tensorized hyperpixel, called t-hyperpixel in this paper, is represented by a t-vector in R^D, i.e., an array in $\mathbb{R}^{3\times 3\times D}$.

Figure 7 shows the tensorization of a canonical vector extracted from a hyperspectral cube. The tensorization of all vectors yields a tensorized hyperspectral cube in $\mathbb{R}^{3\times 3\times D_1\times D_2\times D}$.

6.3.3 Input Matrices and T-Matrices

To classify a query hyperpixel, it is necessary to extract features from the hyperpixel. A t-hyperpixel in TGCA is represented by a set of t-vectors in the 5×5 neighborhood of the t-hyperpixel. These t-vectors are used to construct a t-matrix. A similar construction is used for GCA.

In GCA for example, let the vectors in the 5×5 neighborhood of a hyperpixel be $X_{\text{vec},1}, \ldots, X_{\text{vec},25}$. The ordering of the vectors should be the same for all hyperpixels. The raw matrix X_{mat} representing the hyperpixel is given by marshalling these vectors as the columns of X_{mat}, namely $X_{\text{mat}} \doteq [X_{\text{vec},1}, \ldots, X_{\text{vec},25}] \in \mathbb{R}^{D\times 25}$. The associated t-matrix $X_{\text{TM}} \in C^{D\times 25}$ in TGCA is obtained by marshalling the associated 25 t-vectors.

After obtaining each matrix and t-matrix, the columns are orthogonalized. The resulting matrices and t-matrices are input samples for GCA and TGCA, respectively.

6.3.4 Classification

To evaluate GCA, TGCA and the competing methods, the overall accuracies (OA) and the Cohen's κ indices of the supervised classification of hyperpixels (i.e., prediction of class labels of hyperpixels) are used. The overall accuracies and κ indices are obtained for different component analyzers and classifiers. Higher values of OA or κ indicate a higher component analyzer performance [9]. Let K be the number of query samples, and let K' be the number of correctly classified samples. The overall accuracy is simply defined by $OA = K'/K$. The κ index is defined by [5]

$$\kappa = \frac{K \cdot K' - \sum_{j=1}^{N^{\text{class}}} a_j b_j}{K^2 - \sum_{j=1}^{N^{\text{class}}} a_j b_j}, \tag{47}$$

where N^{class} is the number of classes, a_j is the number of samples belonging to the jth class and b_j is the number of samples classified to the jth class.

Two classical component analyzers, namely PCA and LDA, and four state-of-the-art component analyzers, namely TDLA [40], LTDA [42], GCA [13] and TPCA (ours), are evaluated against TGCA. As an evaluation baseline, the results obtained with the original raw canonical vectors for hyperpixels are given. These raw vectors are denoted as the "original" (ORI for short) vectors. Three vector-oriented classifiers, NN (nearest neighbor), SVM (support vector machine) and RF (random forest), are employed to evaluate the effectiveness of the features extracted by these component analyzers.

In the experiments, the background hyperpixels are excluded, because they do not have labels in the ground truth. A total of 10% of the foreground hyperpixels are randomly and uniformly chosen without replacement as the observed samples (i.e., samples whose class labels are known in advance). The rest of the foreground hyperpixels are chosen as the query samples, that is, samples with the class labels to be determined.

In order to use the vector-oriented classifiers NN, SVM and RF, the t-vector results, generated by TGCA or TPCA, are transformed by pooling them to yield canonical vectors. For TGCA, the canonical vectors obtained by pooling are referred to as TGCA-I features and the t-vectors without pooling are referred to as the TGCA-II features.

To assess the effectiveness of the TGCA-II features, a generalized classifier which deals with t-vectors is needed. It is possible to generalize many canonical classifiers from vector-oriented to t-vector-oriented; however, a comprehensive discussion of these generalizations is outside the scope of this paper. Nevertheless, it is very straightforward to generalize NN. The d-dimensional t-vectors are not only elements of the module C^d, but also the elements in the vector space $\mathbb{C}^{3\times 3\times d}$. This enables the use of the canonical Frobenius norm to measure the distance between two t-vectors, as the elements in $\mathbb{C}^{3\times 3\times d}$. The canonical Frobenius norm should not be confused with the generalized Frobenius norm defined in Eq. (20).

Figure 8 shows the highest classification accuracies obtained by each pair of component analyzer and classifier on the two hyperspectral cubes. The highest accuracies are obtained by traversing the set of feature dimensions $d \in \{5, 10, \ldots, D_m\}$, where D_m is the maximum dimension valid for the associated component analyzer. Figure 8 shows that the results obtained by the algorithms TPCA, TGCA-I and TGCA-II are consistently better than those obtained by their canonical counterparts. Even working with a relatively weak classifier NN, TGCA achieves the highest accuracies and highest κ indices in the experiments. Further results are shown in Figs. 9 and 10. It is clear that the pair TGCA and NN yield the best results, outperforming any other pair of analyzer and classifier.

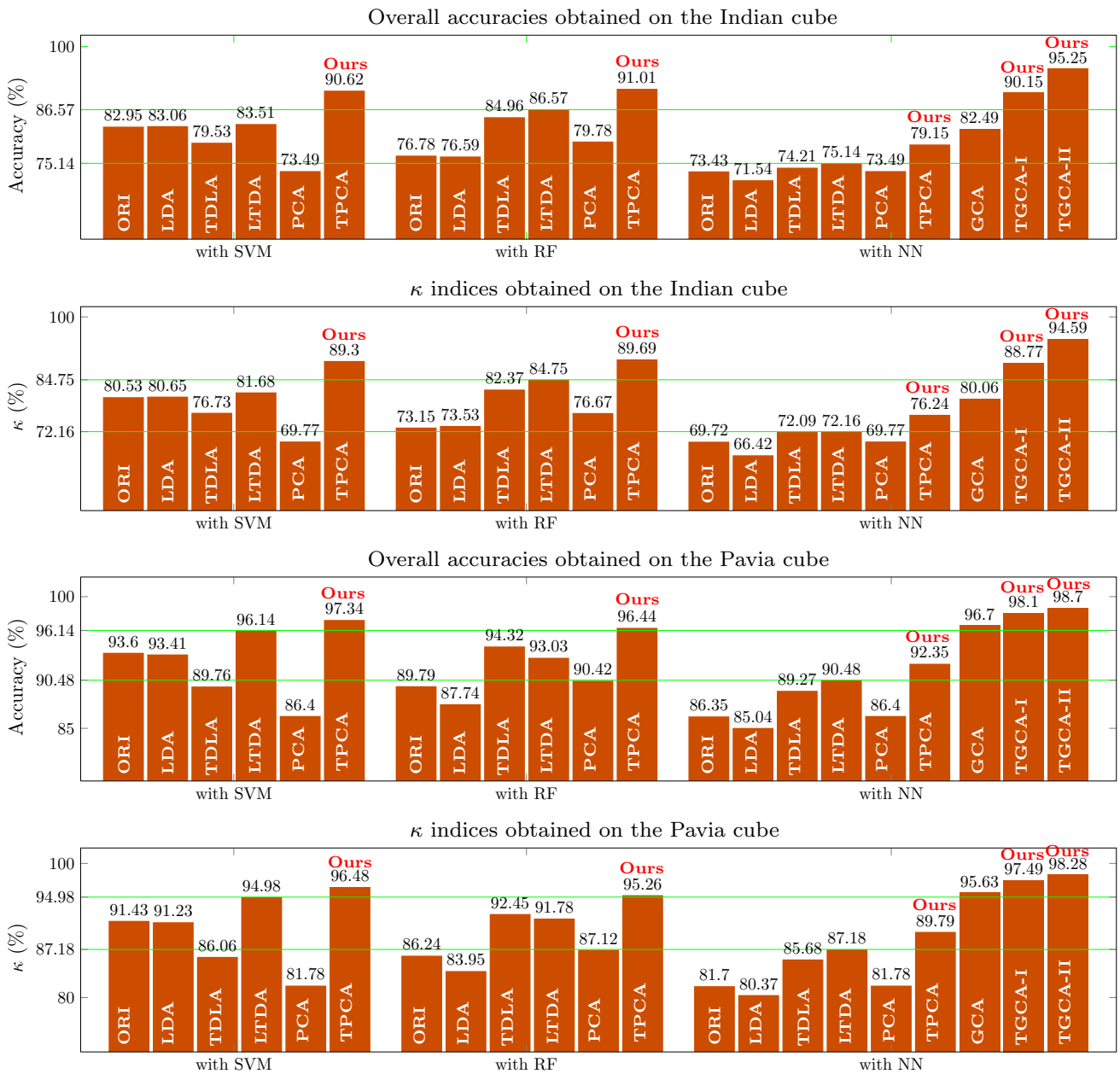


Fig. 8 Classification accuracies obtained on two hyperspectral cubes

6.3.5 TGCA Versus GCA

It is noted that the maximum dimension of the TGCA and GCA features is equal to the number of observed training samples and therefore is much higher than the original dimension, which is equal to the number of spectral bands. Thus, taking the original dimension as the baseline, one can employ TGCA or GCA either for dimension reduction or dimension increase. When the so-called curse of dimension is the concern, one can discard the insignificant entries of the TGCA and GCA features. When the accuracy is the primary concern, one can use higher-dimensional features.

The performances of TGCA and GCA for varying feature dimension are compared using accuracy curves generated by TGCA (i.e., TGCA-I and TGCA-II) and GCA, as shown in Fig. 11. The results are obtained for low feature dimensions and for high feature dimensions. It is clear that the classification accuracies obtained using TGCA and TGCA-II are consistently higher than the accuracies obtained using GCA.

6.3.6 TPCA Versus PCA

The classification accuracies of TPCA and PCA are compared, although the highest classification accuracies are not

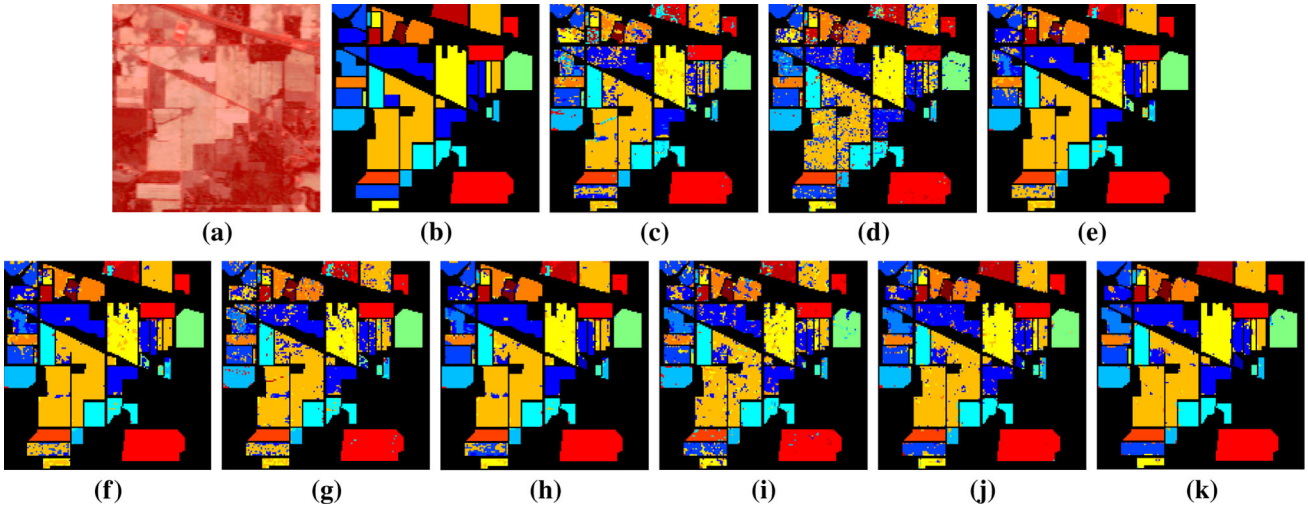


Fig. 9 Some visual results obtained on the Indian cube. **a** Pseudo-colored 2D scene of Indian Pines **b** class ground truth of hyperpixels **c** ORI with RF **d** LDA with RF **e** TDLA with RF **f** LTDA with RF **g** PCA with RF **h** TPCA with RF **i** GCA with NN **j** TGCA-I with NN **k** TGCA-II with NN (Color figure online)

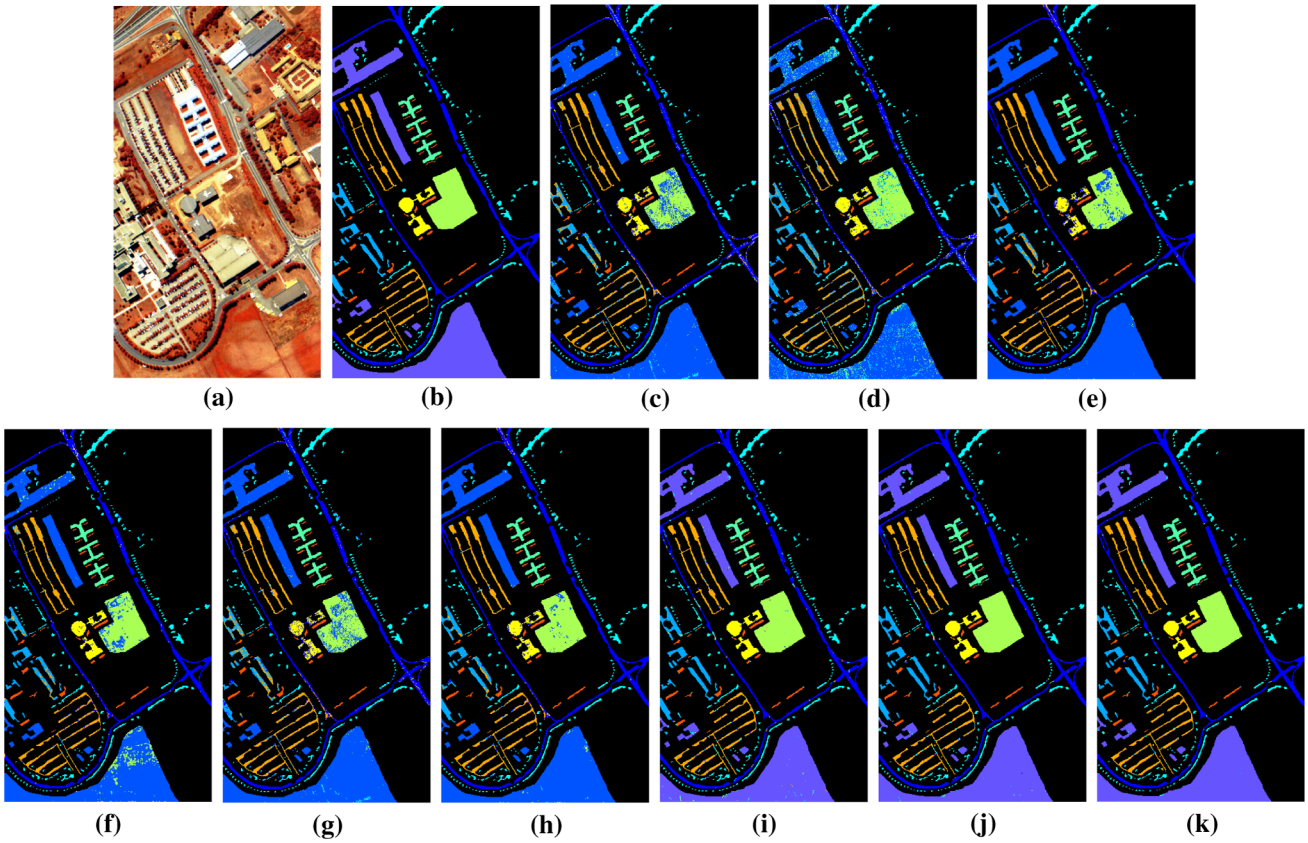


Fig. 10 Some visual results obtained on the Pavia cube. **a** Pseudo-colored 2D scene of the Pavia University **b** class ground truth of hyperpixels **c** ORI with RF **d** LDA with RF **e** TDLA with RF **f** LTDA with RF **g** PCA with RF **h** TPCA with RF **i** GCA with NN **j** TGCA-I with NN **k** TGCA-II with NN (Color figure online)

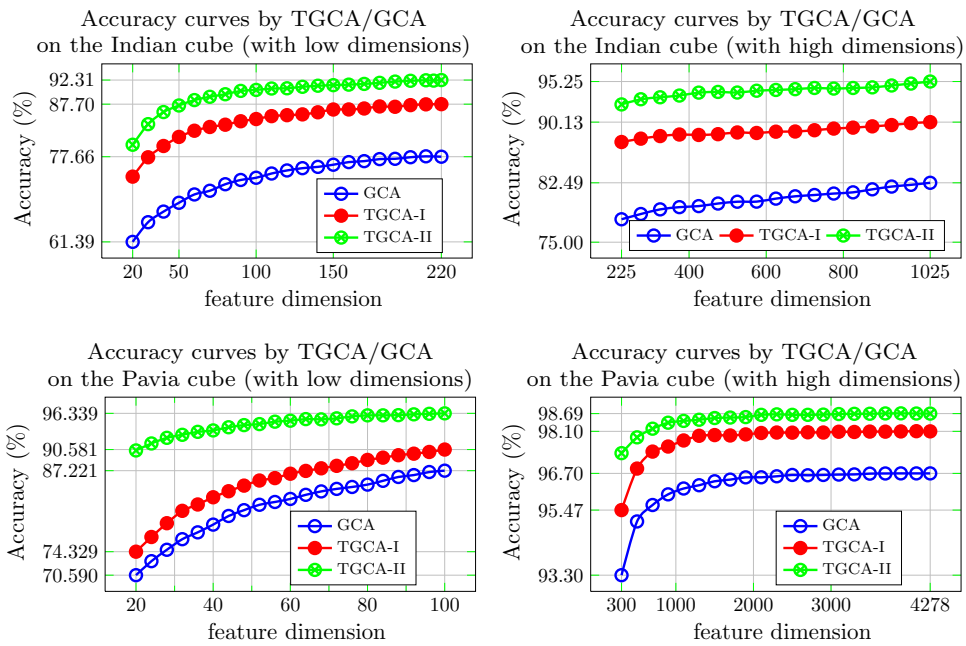


Fig. 11 Accuracy curves obtained by TGCA/GCA (with NN) on the Indian/Pavia cube

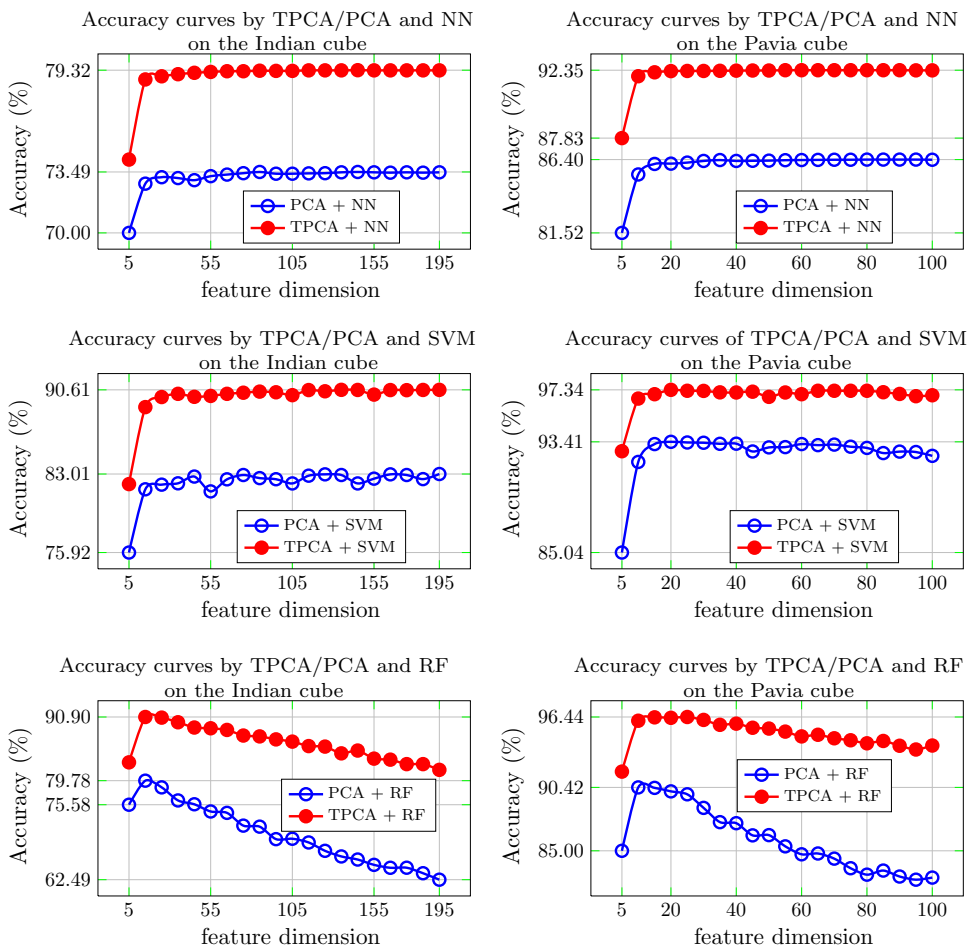
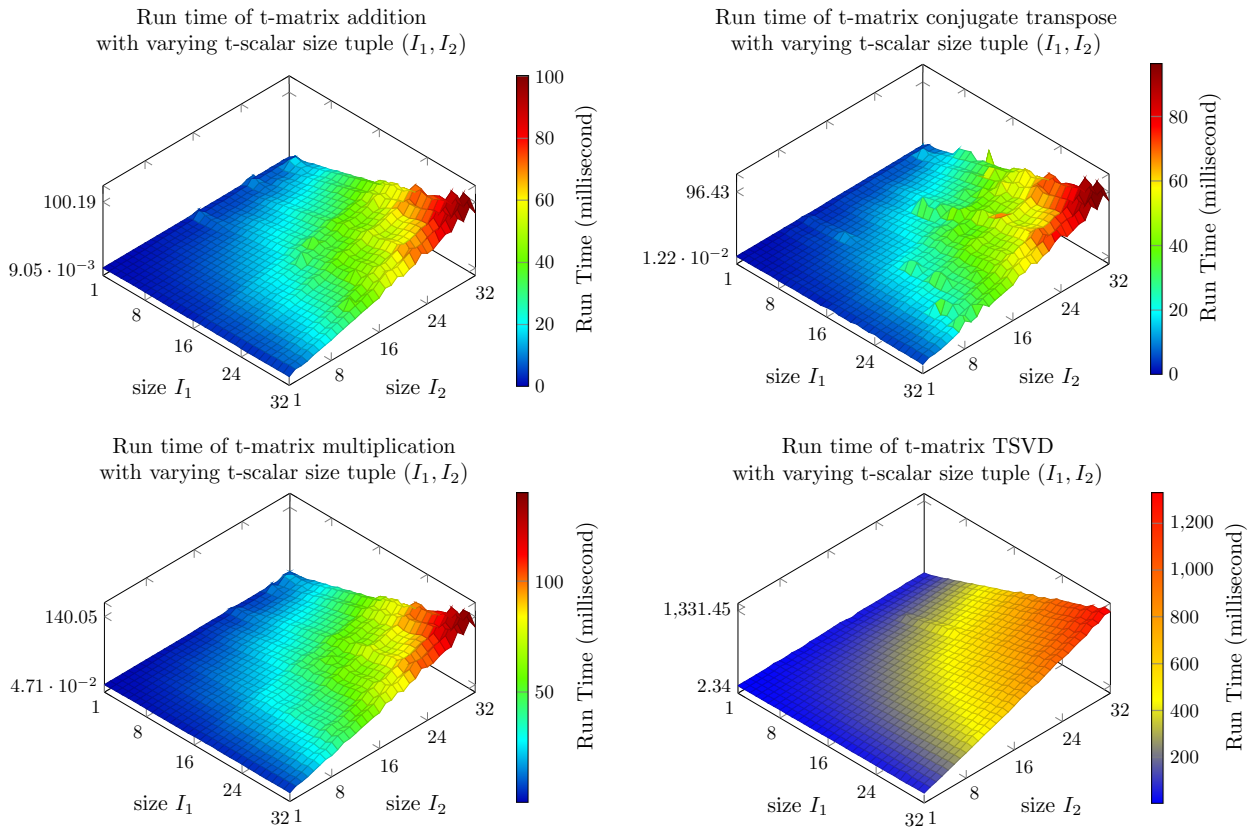


Fig. 12 Accuracy curves obtained by TPCA/PCA and different classifiers on the Indian/Pavia cube. First column: results on the Indian cube. Second column: results on the Pavia cube



Run time (milliseconds) of t-matrix addition/conjugate transposition/multiplication/TSVD with some different t-scalar size duples

(I_1, I_2)	$(1,1)$	$(4,4)$	$(8,8)$	$(12,12)$	$(16,16)$	$(20,20)$	$(24,24)$	$(28,28)$	$(32,32)$
addition	0.009	1.278	4.371	11.075	19.906	32.610	51.206	67.237	83.592
conjugate transpose	0.012	1.434	4.462	11.297	19.630	33.012	51.696	65.216	82.747
multiplication	0.047	1.846	6.730	15.786	28.253	59.813	72.874	95.064	118.535
TSVD	2.341	18.837	77.587	177.804	321.547	503.543	740.465	1 002.679	1 314.483

† When $(I_1, I_2) = (1, 1)$, the manipulation is equivalent to a canonical matrix manipulation and the reported time does not include the time spent on the Fourier transform and its inverse transform.

Fig. 13 Run time of some t-matrix manipulations with different t-scalar sizes

obtained from TPCA or PCA. The classification accuracy curves obtained by TPCA and PCA (with classifiers NN, SVM and RF) are shown in Fig. 12. It is clear that, no matter which classifier and feature dimension are chosen, the accuracy using TPCA is consistently higher than the accuracy using PCA.[3]

6.4 Computational Cost

The run times of t-matrix manipulations with different t-scalar sizes $I_1 \times I_2$ are shown in Fig. 13. The size of t-scalars ranges from $1 \leq I_1, I_2 \leq 32$. The evaluated t-matrix manipulations include addition, conjugate transposition, multiplication and TSVD. The run time is evaluated using MATLAB R2018B on a notebook PC with Intel i7-4700MQ CPU at 2.40 GHz and 16 GB memory.

Each time point in the figure is obtained by averaging 100 manipulations on random t-matrices in $\mathbb{R}^{I_1 \times I_2 \times 64 \times 64}$. Each t-matrix with $(I_1, I_2) \neq (1, 1)$ is transformed to the Fourier domain and manipulated via its $I_1 \cdot I_2$ slices. The results are transferred back to the original domain by the inverse Fourier transform. Note that when $(I_1, I_2) = (1, 1)$, a t-matrix manipulation is reduced to canonical matrix manipulation. The reported run time of a canonical matrix manipulation does not include the time spent on the Fourier transform and its inverse transform.

[3] To use the same classifiers, pooling is used to transform the t-vectors by TPCA to canonical vectors.

From Fig. 13, it can be seen that the run time is essentially an increasing linear function of the number of slices, i.e., $I_1 \cdot I_2$.

7 Conclusion

An algebraic framework of tensorial matrices is proposed for generalized visual information analysis. The algebraic framework generalizes the canonical matrix algebra, combining the "multi-way" merits of high-order arrays and the "two-way" intuition of matrices. In the algebraic framework, scalars are extended to t-scalars, which are implemented as high-order numerical arrays of a fixed size. With appropriate operations, the t-scalars are trinitarian in the following sense. (1) T-scalars are generalized complex numbers. (2) T-scalars are elements of an algebraic ring. (3) T-scalars are elements of a linear space.

Tensorial matrices, called t-matrices, are constructed with t-scalar elements. The resulting t-matrix algebra is backward compatible with the canonical matrix algebra. Using this t-algebra framework, it is possible to generalize many canonical matrix and vector constructions and algorithms.

To demonstrate the "multi-way" merits and "two-way" matrix intuition of the proposed tensorial algebra and its applications to generalized visual information analysis, the canonical matrix algorithms SVD, HOSVD, PCA, 2DPCA and GCA are generalized. Experiments with low-rank approximation, reconstruction and supervised classification show that the generalized algorithms compare favorably with their canonical counterparts on visual information analysis.

Acknowledgements Liang Liao would like to thank professor Pinzhi Fan (Southwestern Jiaotong University, China) for his support and some insightful suggestions to this work. Liang Liao also would like to thank Yuemei Ren, Chengkai Yang, Haichang Ye, Jie Yang and Xuechun Zhang for their supports to some early stage experiments of this work. All prospective support and collaborations to this research are welcome. Contact email: liaolangis@126.com or liaoliang2018@gmail.com.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Appendix

Before giving a proof of the equivalence of Eqs. (27) and (28), namely the generalized Eckart–Young–Mirsky theorem, some notations need to be defined.

First, rank$(\cdot)$ denotes the rank of a t-matrix, which generalizes the rank of a canonical matrix and is defined as follows.
Definition I, rank of a t-matrix. Given a t-matrix, the rank $Y_T \doteq \text{rank}(X_{\text{TM}})$ is a nonnegative t-scalar such that

$$F(Y_T)_i = \text{rank}(F(X_{\text{TM}})(i)) \geq 0\ ,\quad 1 \leq i \leq I, \tag{48}$$

where $F(X_{\text{TM}})(i)$ denotes the ith slice of the Fourier transform $F(X_{\text{TM}})$.
Definition II, partial ordering of nonnegative t-scalars. Given two nonnegative t-scalars X_T and Y_T, the notation $X_T \leq Y_T$ is equivalent to the following condition:

$$F(X_T)_i \leq F(Y_T)_i\ ,\quad 1 \leq i \leq I. \tag{49}$$

Definition III, minimization of nonnegative t-scalar variable. For a nonnegative t-scalar variable X_T varying in a subset of S^{nonneg}, $Y_T \doteq \min(X_T)$ is the nonnegative t-scalar infimum of the subset, satisfying the following condition:

$$F(Y_T)_i = \min\left(F(X_T)_i\right) \geq 0\ ,\ 1 \leq i \leq I, \tag{50}$$

where $F(Y_T)$ and $F(X_T)$, respectively, denote the Fourier transforms of Y_T and X_T.

Given two nonnegative t-scalars X_T and Y_T, let M_T be the nonnegative t-scalar defined by $M_T = \min(X_T, Y_T)$, namely

$$F(M_T)_i = \min(F(X_T)_i, F(Y_T)_i) \geq 0\ \ ,\ i \leq i \leq I. \tag{51}$$

The above definitions are not casual ones. Following the above definitions, it is not difficult to verify that many generalized rank properties hold in the analogous form of their canonical counterparts.

For examples, given any t-matrices $X_{\text{TM}} \in C^{D_1 \times D_2}$ and $Y_{\text{TM}} \in C^{D_1 \times D_2}$ or $Y_{\text{TM}} \in C^{D_2 \times D_3}$, the following inequalities hold:

$$Z_T \leq \text{rank}(X_{\text{TM}}) \leq \min(D_1, D_2) \cdot E_T\ \ . \tag{52}$$

$$\begin{aligned} Z_T &\leq \text{rank}(X_{\text{TM}} + Y_{\text{TM}}) \\ &\leq \text{rank}(X_{\text{TM}}) + \text{rank}(Y_{\text{TM}})\ \ . \end{aligned} \tag{53}$$

$$\begin{aligned} &\text{rank}(X_{\text{TM}}) + \text{rank}(Y_{\text{TM}}) - D_2 \cdot E_T \\ &\quad \leq \text{rank}(X_{\text{TM}} \circ Y_{\text{TM}}) \\ &\quad \leq \min\left(\text{rank}(X_{\text{TM}}), \text{rank}(Y_{\text{TM}})\right). \end{aligned} \tag{54}$$

Since a t-scalar is a t-matrix of one row and one column, the rank of a t-scalar can be obtained.

Given any t-scalar X_T, let $G_T \doteq \text{rank}(X_T)$ be the rank of X_T. Then, following Eq. (48), it is not difficult to prove the

ith entry of the Fourier transform $F(G_T)$ which is given as follows:

$$F(G_T)_i = \begin{cases} 1, & \text{if } X_{T,i} \neq 0 \\ 0, & \text{otherwise} \end{cases}, \quad 1 \leq i \leq I. \tag{55}$$

Following the partial ordering given as in (49) and Eq. (48), it is not difficult to prove that the following propositions hold:

$$\begin{aligned} &Z_T \leq \text{rank}(X_T) \leq E_T \ , \text{ for all t-scalars} \\ &Z_T = \text{rank}(X_T) \text{ iff } X_T = Z_T \\ &E_T = \text{rank}(X_T) \text{ iff } X_T \text{ is invertible.} \end{aligned} \tag{56}$$

It follows from (56) that $Z_T < \text{rank}(X_T) < E_T$ iff the t-scalar X_T is nonzero and non-invertible.[4]

Generalized rank from a TSVD perspective. Given any t-matrix $X_{\text{TM}} = U_{\text{TM}} \circ S_{\text{TM}} \circ V_{\text{TM}}^{\mathcal{H}}$ where $S_{\text{TM}} \doteq \text{diag}(\lambda_{T,1}, \ldots, \lambda_{T,k}, \ldots \lambda_{T,Q})$ and $\lambda_{T,k} \in C$ is a t-scalar for all k, then the following equation holds and generalizes its canonical counterpart:

$$Z_T \leq \text{rank}(X_{\text{TM}}) \equiv \textstyle\sum_{k=1}^{Q} \text{rank}(\lambda_{T,k}) \leq Q \cdot E_T \ . \tag{57}$$

Let the approximation be $\hat{X}_{\text{TM}} = U_{\text{TM}} \circ \hat{S}_{\text{TM}} \circ V_{\text{TM}}^{\mathcal{H}}$ where $\hat{S}_{\text{TM}} = \text{diag}(\lambda_{T,1}, \ldots, \lambda_{T,r}, \underbrace{Z_T, \ldots, Z_T}_{Q-r})$. Then, $\hat{X}_{\text{TM}}$ is a low-rank approximation to X_{TM} since the following rank inequality holds:

$$\text{rank}(\hat{X}_{\text{TM}}) \equiv \textstyle\sum_{k=1}^{r} \text{rank}(\lambda_{T,k}) \leq \text{rank}(X_{\text{TM}}) \ . \tag{58}$$

Furthermore, it is not difficult to verify that Eq. (28) is equivalent to the following equation in the form of canonical matrices (i.e., slices of Fourier-transformed t-matrices).

$$\begin{aligned} \tilde{X}_{\text{TM}}^{\text{approx}}(i) = \text{argmin}_{Y_{\text{mat}} \in \mathbb{C}^{D_1 \times D_2}} \|\tilde{X}_{\text{TM}}(i) - Y_{\text{mat}}\|_F \\ \text{subject to rank}(Y_{\text{mat}}) \leq r \ , \ 1 \leq i \leq I, \end{aligned} \tag{59}$$

where $\tilde{X}_{\text{TM}}^{\text{approx}}$ and $\tilde{X}_{\text{TM}}$, respectively, denote the Fourier transform of $X_{\text{TM}}^{\text{approx}}$ and X_{TM} in Eq. (28), and $\text{rank}(Y_{\text{mat}})$ is the rank of a complex matrix Y_{mat} in $\mathbb{C}^{D_1 \times D_2}$.

On the other hand, by applying the Fourier transforms to both sides of Eq. (27), Eq. (27) is transformed to the following equation in the form of canonical matrices (i.e., slices of Fourier-transformed t-matrices):

$$\tilde{X}_{\text{TM}}^{\text{svd}}(i) = \tilde{U}_{\text{TM}}(i) \cdot \tilde{S}_{\text{TM}}^{\text{approx}}(i) \cdot \left(\tilde{V}_{\text{TM}}(i)\right)^H, \ 1 \leq i \leq I, \tag{60}$$

where $\tilde{X}_{\text{TM}}^{\text{svd}}$, $\tilde{U}_{\text{TM}}$, $\tilde{S}_{\text{TM}}^{\text{approx}}$ and $\tilde{V}_{\text{TM}}^{\text{approx}}$, respectively, denote the Fourier transforms of $\hat{X}_{\text{TM}}$, U_{TM}, $\hat{S}_{\text{TM}}$ and V_{TM} in Eq. (27).

The canonical Eckart–Young–Mirsky theorem guarantees the equivalence of Eqs. (59) and (60).

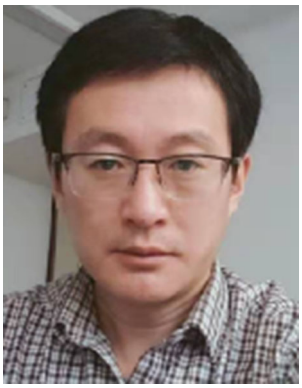
[4] The partial order "<" is defined between nonnegative t-scalars. The inequality $Z_T < \text{rank}(X_T) < E_T$ means $Z_T \leq \text{rank}(X_T) \leq E_T$ and $\text{rank}(X_T) \neq Z_T$ and $\text{rank}(X_T) \neq E_T$.

References

1. Almohammad, A., Ghinea, G.: Stego image quality and the reliability of PSNR. In: 2010 2nd International Conference on Image Processing Theory, Tools and Applications, pp. 215–220 (2010)
2. Bracewell, R.N., Bracewell, R.N.: The Fourier Transform and Its Applications, 3rd edn, pp. 108–112. McGraw-Hill, New York (1999)
3. Braman, K.: Third-order tensors as linear operators on a space of matrices. Linear Algebra Appl. **433**(7), 1241–1253 (2010)
4. Chen, Z., Wang, B., Niu, Y., Xia, W., Zhang, J.Q., Hu, B.: Change detection for hyperspectral images based on tensor analysis. In: Geoscience and Remote Sensing Symposium, pp. 1662–1665 (2015)
5. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Meas. **20**(1), 37–46 (1960)
6. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**(4), 1253–1278 (2000)
7. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika **1**(3), 211–218 (1936)
8. Fan, H., Li, C., Guo, Y., Kuang, G., Ma, J.: Spatial-spectral total variation regularized low-rank tensor decomposition for hyperspectral image denoising. IEEE Trans. Geosci. Remote Sens. **56**(10), 6196–6213 (2018)
9. Fitzgerald, R.W., Lees, B.G.: Assessing the classification accuracy of multisource remote sensing data. Remote Sens. Environ. **47**(3), 362–368 (1994)
10. Fu, W., Li, S., Fang, L., Kang, X., Benediktsson, J.A.: Hyperspectral image classification via shape-adaptive joint sparse representation. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. **9**(2), 556–567 (2016)
11. Gleich, D.F., Chen, G., Varah, J.M.: The power and Arnoldi methods in an algebra of circulants. Numer. Linear Algebra Appl. **20**(5), 809–831 (2013)
12. Golub, G., Loan, C.V.: Matrix Computations, Chap. 2. North Oxford Academic, Oxford (1983)
13. Harandi, M., Hartley, R., Shen, C., Lovell, B., Sanderson, C.: Extrinsic methods for coding and dictionary learning on Grassmann manifolds. Int. J. Comput. Vis. **114**(2), 113–136 (2015)
14. Harandi, M.T., Hartley, R., Lovell, B., Sanderson, C.: Sparse coding on symmetric positive definite manifolds using Bregman divergences. IEEE Trans. Neural Netw. Learn. Syst. **27**(6), 1294–1306 (2015)
15. He, Z., Li, J., Liu, L.: Tensor block-sparsity based representation for spectral-spatial hyperspectral image classification. Remote Sens. **8**(8), 636 (2016)
16. Hungerford, T.: Algebra, Graduate Texts in Mathematics, Chap. IV, vol. 73. Springer, New York (1974)
17. Kilmer, M.E., Braman, K., Hao, N., Hoover, R.C.: Third-order tensors as operators on matrices: a theoretical and computational

framework with applications in imaging. SIAM J. Matrix Anal. Appl. **34**(1), 148–172 (2013)
18. Kilmer, M.E., Martin, C.D.: Factorization strategies for third-order tensors. Linear Algebra Appl. **435**(3), 641–658 (2011)
19. Kolda, T., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
20. Liao, L., Maybank, S.J., Zhang, Y., Liu, X.: Supervised classification via constrained subspace and tensor sparse representation. In: International Joint Conference on Neural Networks, pp. 2306–2313 (2017)
21. Liu, Z., Tang, B., He, X., Qiu, Q., Wang, H.: Sparse tensor-based dimensionality reduction for hyperspectral spectral-spatial discriminant feature extraction. IEEE Geosci. Remote Sens. Lett. **1775–1779**(99), 1–5 (2017)
22. Lu, C., Feng, Y., Liu, W., Lin, Z., Yan, S.: Tensor robust principal component analysis: exact recovery of corrupted low rank tensors via convex optimization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5249–5257 (2016)
23. Lu, H., Plataniotis, K.N., Venetsanopoulos, A.N.: MPCA: multilinear principal component analysis of tensor objects. IEEE Trans. Neural Netw. **19**(1), 18–39 (2008)
24. Ma, X., Wang, H., Geng, J.: Spectral-spatial classification of hyperspectral image based on deep auto-encoder. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. **9**(9), 4073–4085 (2016)
25. Muralidhara, C., Gross, A.M., Gutell, R.R., Alter, O.: Tensor decomposition reveals concurrent evolutionary convergences and divergences and correlations with structural motifs in ribosomal RNA. PloS One **6**(4), e18768 (2011)
26. Omberg, L., Golub, G.H., Alter, O.: A tensor higher-order singular value decomposition for integrative analysis of DNA microarray data from different studies. Proc. Natl. Acad. Sci. U S A **104**, 18371–18376 (2007)
27. Papalexakis, N.S.L.D.X.F.K.H.E., Faloutsos, C.: Tensor decomposition for signal processing and machine learning. IEEE Trans. Signal Process. **65**(13), 3551–3582 (2017)
28. Ren, Y., Liao, L., Maybank, S.J., Zhang, Y., Liu, X.: Hyperspectral image spectral-spatial feature extraction via tensor principal component analysis. IEEE Geosci. Remote Sens. Lett. **14**(9), 1431–1435 (2017)
29. Taguchi, Y.H.: Tensor decomposition-based unsupervised feature extraction applied to matrix products for multi-view data processing. Plos One **12**(8), e0183933 (2017)
30. Tucker, L.R.: Some mathematical notes on three-mode factor analysis. Psychometrika **31**(3), 279–311 (1966)
31. University of the Basque country: hyperspectral remote sensing scenes. http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes
32. Vannieuwenhoven, N., Vandebril, R., Meerbergen, K.: A new truncation strategy for the higher-order singular value decomposition. SIAM J. Sci. Comput. **34**(2), A1027–A1052 (2012)
33. Vasilescu, M.A.O.: Human motion signatures: analysis, synthesis, recognition. Proc. Int. Conf. Pattern Recognit. **3**, 456–460 (2002)
34. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: TensorFaces. In: European Conference on Computer Vision, pp. 447–460. Springer (2002)
35. Vasilescu, M.A.O., Terzopoulos, D.: TensorTextures: multilinear image-based rendering. ACM Trans. Graph. **23**(3), 336–342 (2004)
36. Wei, Y., Zhou, Y., Li, H.: Spectral-spatial response for hyperspectral image classification. Remote Sens. **9**(3), 203–233 (2017)
37. Yang, J., Zhang, D., Frangi, A.F., Yang, J.Y.: Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE Trans. Pattern Anal. Mach. Intell. **26**(1), 131–7 (2004)
38. Zhang, E., Zhang, X., Jiao, L., Li, L., Hou, B.: Spectral-spatial hyperspectral image ensemble classification via joint sparse representation. Pattern Recognit. **59**, 42–54 (2016)
39. Zhang, J., Saibaba, A.K., Kilmer, M., Aeron, S.: A randomized tensor singular value decomposition based on the t-product. Numer. Linear Algebra Appl. **25**(5), e2179 (2018)
40. Zhang, L., Zhang, L., Tao, D., Huang, X.: Tensor discriminative locality alignment for hyperspectral image spectral-spatial feature extraction. IEEE Trans. Geosci. Remote Sens. **51**(1), 242–256 (2013)
41. Zhang, Z., Ely, G., Aeron, S., Hao, N., Kilmer, M.: Novel methods for multilinear data completion and de-noising based on tensor-SVD. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3842–3849 (2014)
42. Zhong, Z., Fan, B., Duan, J., Wang, L., Ding, K., Xiang, S., Pan, C.: Discriminant tensor spectral-spatial feature extraction for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. **12**(5), 1028–1032 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Liang Liao received his B.S., M.S. and Ph.D. degrees, respectively, from Northwestern Polytechnical University, Southwest Jiaotong University and South China University of Technology, all in Electronics and Information Technology. He is also an associate professor and the associate dean of the Department of Electronics and Information Engineering, School of Electronics and Information, Zhongyuan University of Technology, Zhengzhou, China. His current interests are machine learning, image recognition and generalized visual information analysis.

Stephen John Maybank is a professor with the Department of Computer Science and Information Systems, Birkbeck College, University of London, UK. He is a Fellow of the IEEE, Fellow of the Royal Statistical Society, and Fellow of the Higher Education Academy, and a recipient of the Koenderink Prize Award at the 10th European Conference on Computer Vision.