ORIGINAL RESEARCH PAPER

Tensor dimensionality reduction via mode product and HSIC

Guo Niu1 | Zhengming Ma2

1 School of Electronics and Information Technology, Foshan University, Foshan, China
2 School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China

Correspondence
Zhengming Ma, School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China.
Email: issmzm@mail.sysu.edu.cn

Funding information
Natural Science Foundation of China, Grant/Award Number: 62006042; Guangdong Basic and Applied Basic Research Foundation, Grant/Award Number: 2019A1515111143

Abstract
Tensor dimensionality reduction (TDR) is a hot research topic in machine learning, which learns data representations by preserving the original data structure while avoiding convert samples into vectors and solving the problem of the curse of dimensionality of tensor data. In the work, a novel TDR approach based on mode product and Hilbert–Schmidt Independence criterion (HSIC) is proposed. The contributions of authors’ work is described as following: (1) HSIC measures the statistical correlation of two random variables. However, instead of measuring the statistical correlation of two random variables directly, HSIC first transforms the two random variables into two reproducing kernel Hilbert spaces (RKHSs), and then measures the statistical correlation of transformed random variables by using Hilbert–Schmidt operators between the two RKHSs. The exploitation of RKHS increases the flexibility and applicability of HSIC. Although HSIC is widely used in machine learning, the authors have not seen its application to dimensionality reduction (DR)(except for authors’ previous work). (2) A novel HSIC-based TDR approach is proposed, which first applies HSIC to capture statistical information of tensor data set for DR. The authors give the mathematical derivation of HSIC for tensor data and establish a framework of TDR based on HSIC, named HSIC-TDR for short, which aims to improve the DR results of tensor by exploring and preserving the statistical information of original data set. (3) Furthermore, to solve the out-of-sample problem, the authors learn an explicit expression between the dimensionality-reduced tensors and the higher-dimensional tensors by introducing mode product to HSIC-TDR. The experimental results between the proposed method and other state-of-the-art algorithm on various datasets demonstrate the well performance of the proposed method.

1 | INTRODUCTION

With the progress of powerful sensors and computer technologies, unprecedented massive data (multi-dimensional data) are exponentially grown in many areas, such as Internet [1], geophysical data [2] and human genome [3] and so on. The processing technologies of these data have become a hot research topic, especially, dimensionality reduction (DR) is one of the most important technology [4]. In general, most DR methods need to convert the multi-dimensional data into the high-dimensional vector before preforming their algorithms. Obviously, the process of the transformation ignores the underlying structure of original data sets and increases a huge computational cost. Therefore, the researches turn to focus on the directly dimensional reduction of the multi-dimensional data (tensor data).

Tensor data represents multi-dimensional data, in which each dimension contains amount of information of the original data, avoiding image-to-vector transformation. Particularly, the matrix or 2-D data can be regarded as a special data of tensor [5, 6] (the 2-D methods only suitable for 2-D images). Tensor-based methods have well performance in many application fields, such as image analysis [5, 7, 8], recommendation system [9, 10], wireless spectrogram generation [11], seismic signals processing [12], computer vision [13], hyperspectral image analysis [14–16], sensor signal processing [17] and so on. Recently, many tensor-based dimensionality reduction (TDR) algorithms have been proposed. Some proposed TDR
algorithms are inspired by famous vector-based DR, such as multi-linear PCA (MPCA) [18], tensor maximum margin criterion (TMMC) [19], multi-linear discriminant analysis (MLDA) [20], tensor discriminant locality alignment (TDLA) [21], tensor locality preserving projection (TLLPP) [22] and tensor marginal fisher analysis (TMFA) [23]. Manifold learning, as a hot topic in non-linear DR area, also is introduced to TDR. For example, Zhang et al. employed the neighbourhood relationship between data to learn the low-rank factorization model, named low-rank matrix decomposition with manifold regularization (MMF) [24]; graph-Laplace tensor decomposition algorithm (GLTD) [25] applies the properties of the graph and similar information to improve the robustness of the algorithm; [26] employed local homeomorphism of manifold learning algorithm and subspace learning for the DR of tensors. Some other researchers focus on tensor decomposition for TDR. Such as Zhou et al. [27] proposed an effective non-negative Tucker decomposition (NTD) to approximate the tensor by the mode product of a low-dimensional tensor with effective projection matrices; Zhang et al. [28] used the low-rank regularized heterogeneous tensor decomposition (LRR-HTD) to learn a set of orthogonal factor matrices for data; Cai et al. [29] proposed a regularized non-negative matrix factorization (GNMF) algorithm, which introduced an affinity graph to obtain the geometric information for seeking a factor matrix factorization; [30] applied sparse Tucker tensor decomposition to the compression of the hyperspectral image or video sequence. These tensor-based models have well performance in TDR. But they merely consider the statistical information of tensor data, i.e. without taking the essential global structure of the tensor data into account. This paper introduces the statistical information of tensor data to TDR, and proposes a novel DR algorithm based on Hilbert–Schmidt independence criterion (HSIC). HSIC [31, 32] measures the statistical correlation between two random vector sets. But instead of directly measuring the statistical correlation of two data sets, HSIC converts the two random vector sets into two regenerative reproducing kernel Hilbert spaces (RKHSs), and then employ the Hilbert–Schmidt (HS) operators of the two RKHSs to measure their statistical correlation [31]. HSIC has been applied in machine learning, such as feature selection [33–35], domain subspace learning [36], hyperspectral image classification [37] and so on. Generally, the sizes of two data sets in HSIC are assumed to be equal. However, this constraint usually is not to be meet in practice. Refs. [38] and [39] addressed the problem by the so-called surrogate kernel matrices. When HSIC is applied to feature selection [34, 35], it actually measures the statistical correlation between the column vectors of data set matrix (represent feature vectors whose components are to be selected) and the corresponding labels. Gangeh et al. introduced HSIC to supervised dictionary learning [40, 41] by first mapping original data set to subspace and then exploring the statistical correlation between mapped data and the corresponding labels. Although HSIC is widely used in machine learning, we have not seen its application to DR (except for our previous work [42]). But same to vector-based DR approaches, these methods cannot directly solve the tensor data. This paper proposes a novel DR method for tensor data based on mode product and HSIC, which has the merits of the HSIC and the mode product. The proposed algorithm has the following contributions.

1. A novel HSIC-based TDR approach is first proposed to capture statistical information of tensor data set. HSIC gains success in many areas of machine learning since it is proposed. However, we have not seen the reported work about HSIC-based DR, expect for our previous work [42]. Although our team has proved the effectiveness of HSIC for DR, it is only suitable for vector data, cannot be directly applied to tensor data. Thus, the first contribution in the paper is to provide the theoretical description of HSIC for TDR.

2. Based on the theory of HSIC, tensor data is transformed to RKHS by the reproducing kernel of RKHS. Therefore, the flexibility and applicability of HSIC can be increased by choosing different RKHS. Moreover, with it, HSIC-TDR becomes a non-linear DR algorithm. The experimental results prove the effectiveness of our proposed method in TDR.

3. In order to solve the out-of-sample problem, a mode product-based HSIC-TDR algorithm (MPHSIC-TDR) is proposed to offer an explicit expression between low-dimensional representation and high-dimensional tensor data. In tensor algebra, the mode product of tensor and matrix is used to change the dimensionality of tensor, and thus we assume the reduced tensor is the mode product of original tensor and a series of matrices. Therefore, MPHSIC-TDR optimizes these matrices by maximizing HSIC.

The remaining part of the paper is organized as follows. In Section 2, we describe the tensor algebra for better understanding the tensor representation and tensor decomposition, and introduce the basic definition of RKHS and HSIC. Section 3 gives some related work about TDR algorithms for comparison. Section 4 contains two part, the framework of TDR with HSIC, and the proposed algorithm (MPHSIC-TDR). Experimental results are presented in Section 5, including data sets description, clustering experiments and classification experiments, and experiment analysis. Finally, Section 6 gives a conclusion of our work.

2 | PRELIMINARIES

2.1 | Tensor algebra

The tensor \( \mathcal{X} \) represents a multi-dimensional data set, \( \mathcal{X} = \{ x_{i_1 \cdots i_n} | x_{i_1 \cdots i_n} \in \mathbb{R}, 1 \leq i_1 \leq L_{x_1}, \ldots, n \in 1, \ldots, N \} \). \( N \) is the order of \( \mathcal{X} \), and \( L_{x_n} \) is the \( n \)th dimension of the tensor data. The tensor \( \mathcal{X} \) can be simply written as \( \mathcal{X} \in \mathbb{R}^{L_{x_1} \times \cdots \times L_{x_N}} \). Obviously, the number of data contained in the tensor \( \mathcal{X} \) is \( |\mathcal{X}| = \prod_{\gamma=1}^{N} L_{x_\gamma} \).
Tensors can be expressed in the form of matrix. For example, the following representation is called the $n$ mode of tensor $\mathcal{X}$

$$
\mathcal{X}^{(n)} = 
\begin{bmatrix}
    x_{1} & 1 & 1 & \cdots & 1 & x_{L_{n+1}} & \cdots & x_{L_{N-n}} \\
    \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
    \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
    x_{1} & 1 & 1 & \cdots & 1 & x_{L_{n+1}} & \cdots & x_{L_{N-n}} \\
\end{bmatrix}_{N \times N}
$$

and $\mathcal{X}^{(n)} \in R^{l_1 \times (l_2 \cdots l_{n-1} \cdots l_{N-n})}$.

The mode representation of a tensor is completely equivalent to the set representation of tensors, and to know any mode of tensors is to know tensors. The mode product is an important operation in tensor algebra. Let $\mathcal{X} \in R^{l_1 \times \cdots \times l_N}$, $A_\mu \in R^{l_\mu \times l_\nu}$, then the $n$ mode product of $\mathcal{X}$ and $A_\mu$ is a new tensor $\mathcal{G} \in R^{l_1 \times \cdots \times l_n \times \cdots \times l_{N-1}}$, i.e.

$$
\mathcal{G} = \mathcal{X} \times_n A_\mu.
$$

If $\mathcal{G}$ is the low-dimensional representation of $\mathcal{X}$, the task of the DR of tensor $\mathcal{X}$ becomes learn an optimal matrix $A_\mu$. Furthermore, the $n$ mode product of $\mathcal{G}$ can be written as follows:

$$
\mathcal{G}^{(n)} = A_\mu \mathcal{X}^{(n)}.
$$

The operation of the mode product satisfies the following characters:

1. **Associative Law**: If $\mathcal{X} \in R^{l_1 \times l_2 \times \cdots \times l_N}$, $A_\mu \in R^{l_\mu \times l_\nu}$ and $B_\nu \in R^{l_\mu \times l_\nu}$, then

$$
\mathcal{X} \times_n A_\mu \times_n B_\nu = \mathcal{X} \times_n (B_\nu A_\mu).
$$

2. **Commutative Law**: If $\mathcal{X} \in R^{l_1 \times l_2 \times \cdots \times l_N}$, $A_\mu \in R^{l_\mu \times l_\nu}$, $A_\mu \in R^{l_\mu \times l_\nu}$ and $n \neq m$, then

$$
\mathcal{X} \times_n A_\mu \times_n A_\nu = \mathcal{X} \times_m A_\mu \times_m A_\nu.
$$

**Theorem 1.** Suppose $\mathcal{G} \in R^{l_1 \times l_2 \times \cdots \times l_N}$, $A_\mu \in R^{l_\mu \times l_\nu}$, $n = 1, \ldots, N-1$, if

$$
\mathcal{X} = \mathcal{G} \times_1 A_1 \cdots \times_{N-1} A_{N-1},
$$

then

$$
\mathcal{X}^{(N)} = \mathcal{G}^{(N)} (A_1 \otimes \cdots \otimes A_{N-1})^T.
$$

Here, $\otimes$ represents the Kronecker product between two matrices.

### 2.2 Reproducing kernel Hilbert space

HSIC is developed based on the theory of RKHS. To better understand HSIC and our method, we introduce some definition of RKHS. RKHS can be generated from kernel function, or from function space with data. Such as, let $f : \Omega \rightarrow R$ be a mapping function from data space $\Omega$ to real data space $R$, then all the $f$, which meets $\int_{\Omega} |f(x)|^2 \text{d}x < +\infty$, construct a square integrable function space $S(\Omega)$. Furthermore, a complete inner product space (Hilbert space) $H$ can be generated with $S(\Omega)$ and the inner product $(\cdot, \cdot)$ on $S(\Omega)$. For example, for any $f, g \in S(\Omega)$, the inner product of them can be defined as $(f, g) = \int_{\Omega} f(x)g(x)\text{d}x$. If a function $k : \Omega \times \Omega \rightarrow R$ meets that

1. Symmetry: for any $x \in \Omega$, $k(x, x) = k(x, x)$,
2. Positive definiteness: for any finite number of data: $x_1, \ldots, x_N \in \Omega$, the kernel matrix is positive definite,

$$
K = 
\begin{bmatrix}
    k(x_1, x_1) & \cdots & k(x_1, x_N) \\
    \vdots & \ddots & \vdots \\
    k(x_N, x_1) & \cdots & k(x_N, x_N) \\
\end{bmatrix},
$$

then $k$ is the reproducing kernel of $H$, and $(H, k)$ is the corresponding RKHS.

According to the above definition, we also can utilize the kernel function $k$ to generate RKHS. Following Mason's theorem [43], one kernel function only produces one RKHS, and is defined with the following characters:

1. **Symmetry**: for any data $x, y \in \Omega$, $k(x, y) = k(y, x)$,
2. **Positive definiteness**: for any finite number of data: $x_1, \ldots, x_N \in \Omega$, the kernel matrix is positive definite:

$$
K = 
\begin{bmatrix}
    k(x_1, x_1) & \cdots & k(x_1, x_N) \\
    \vdots & \ddots & \vdots \\
    k(x_N, x_1) & \cdots & k(x_N, x_N) \\
\end{bmatrix},
$$

### 2.3 Hilbert-Schmidt independence criterion

HSIC measures the statistical correlation between two random variables $X$ and $Y$. In fact, HSIC does not directly measure the statistical correlation between $X$ and $Y$. It converts two random vector sets into two regenerative RKHSs, and then employs the HS operators of the two RKHSs to measure their statistical correlation. Therefore, the HSIC [31] of two random variables $X$ and $Y$ is defined as

$$
\text{HSIC}(X, Y) = E_{XY} \left[ \| (\varphi_X(X) - \mu_X) \otimes (\varphi_Y(Y) - \mu_Y) \|_{HS}^2 \right],
$$

here, $\| \cdot \|_{HS}$ denotes the norm on HS operators space, $E_{XY}$ denotes the cross-covariance operate of $X$ and $Y$, $\varphi_X(X)$ and $\varphi_Y(Y)$ describe that convert the two random variables into two corresponding RKHSs, $\mu_X$ and $\mu_Y$ are the mean function of $X$ and $Y$ in the corresponding RKHS $H_X$ and $H_Y$, respectively. The HS operators space is defined as follows.
Definition 1. Let $H_X$ and $H_Y$ be two separable Hilbert spaces and $\{ e_i^X \}_{i \in I}$ be the standard orthogonal basis of $H_X$. Let $T : H_X \to H_Y$ be a compact operator, if $\sum_{i \in I} \| T e_i^X \|_Y^2 < +\infty$, then we can call $T$ is the HS operator.

Theorem 2. Let $HS(H_X \to H_Y)$ be the linear space composed of all HS operators from $H_X$ to $H_Y$. For any $T, S \in HS(H_X \to H_Y)$, $\sum_{i \in I} \langle T e_i^X, S e_i^X \rangle_Y < +\infty$, then $(HS(H_X \to H_Y), \langle \cdot, \cdot \rangle_{HS})$ is a Hilbert space, where the inner product $(\cdot, \cdot)_{HS}$ is as follows:

$$\langle T, S \rangle_{HS} = \sum_{i \in I} \langle T e_i^X, S e_i^X \rangle_Y.$$

Theorem 3. Let $H_X$ and $H_Y$ be two separable Hilbert spaces, and $f_0 \in H_X \otimes \mathbb{G}_0 \subset H_Y$, we can determine $f_0 \otimes \mathbb{G}_0 : H_X \to H_Y$ that if for $f \in H_X$,

$$f_0 \otimes \mathbb{G}_0 (f) = (f_0, f)_{H_X \otimes \mathbb{G}_0} \in H_Y \quad (8)$$

then, $f_0 \otimes \mathbb{G}_0 \in HS(H_X \to H_Y)$.

Remark: $f_0 \otimes \mathbb{G}_0$ is called as a tensor product of $f_0$ and $\mathbb{G}_0$, and Theorem 3 shows that, $f_0 \otimes \mathbb{G}_0$ is an HS operator.

The definition of HSIC indicates that HSIC essentially measures of the statistical correlation between $\varphi_X (X)$ and $\varphi_Y (Y)$. In practice, the probability distributions of random variables $X$ and $Y$ are not known. Therefore, according to probability theory, we can use the sample average instead of statistical average, and assume that the probability of random event $\{ X = x_i, Y = y_j \}$ is 0. Then, HSIC (Equation 6) can be derived as the following from [42]

$$HSIC (X, Y) \approx \frac{1}{N^2} \text{tr} (K_Y C_X K_X C_N), \quad (9)$$

here $C_N = I_N - \frac{1}{N} \Gamma_N \Gamma_N^T$ is the centralized matrix, $K_X$ and $K_Y$ are the kernel matrices with data sets $X$ and $Y$, respectively.

3 RELATED WORK

Tensor analysis has been widely used in machine learning [8, 17, 44]. Tensor analysis (TA) [44] and tensor decomposition [45, 46] are two important work for TDR. Ref. [17] gives an overview of tensor-based models and methods for multi-sensor signal processing, such as the Tucker decomposition [27, 47], the canonical polyadic decomposition [48], the tensor-train decomposition (TTD), the structured TTD and so on. Ma et al. [16] used tensor singular value decomposition (t-SVD) to learn a low-rank tensor from limited coefficients in any orthonormal basis. Liu et al. [49] proposed a proximal operator for the approximation of tensor nuclear norms based on tensor-train rank-1 decomposition with the SVD. Here, we describe some related TDR work in detail, especially used in our experiments for comparison.

Many TDR approaches are around the Tucker decomposition of tensor. Let $\mathcal{T} \in R^{I_1 \times \cdots \times I_N}$ be a tensor, the Tucker decomposition of tensor $\mathcal{T}$ can be expressed as follows:

$$\min_{G \in A_1 \cdots A_N} \| \mathcal{T} - G X_1 A_1 \cdots X_N A_N \|_F^2,$$

where $A_n \in R^{I_n \times J_n}, n = 1, ..., N$, and $G \in R^{I_1 \times \cdots \times I_N}$. If $I_n \gg J_n$, then $G$ is the low-dimensional representation of $\mathcal{T}$.

Since the matrices can be regarded as the second-order tensors, the famous matrix non-negative factorization can be regarded as special kind of tensor Tucker composition. Tucker composition can be combined with various regularization terms for TDR. For example, the regularized non-negative matrix factorization algorithm (GNMF) [29] is described as

$$\min_{U \geq 0, Y \geq 0} \| \mathcal{X} - U Y \|_F^2 + \lambda tr (Y \Pi Y^T), \quad (11)$$

where $X \in R^{I \times N}, U \in R^{I \times d}$ and $Y \in R^{N \times N}$. $\Pi \in R^{N \times N}$ is the graph regularized Laplacian matrix, and can be calculated by the similarity between the column vectors of $X$. If the column vectors of $X$ are regarded as the data samples, the column vectors of $Y$ are the compact representation of data, as claimed in [29].

Manifold regularization non-negative Tucker decomposition (MRNTD) [50] transforms the GNMF into the tensor version:

$$\min_{G \geq 0, A_n \geq 0, n = 1, ..., N} \| \mathcal{T} - G X_1 A_1 \cdots X_N A_N \|_F^2$$

$$+ \lambda tr \left( \left( G (N) \right)^T \Pi G (N) \right), \quad (12)$$

where $X \in R^{I_1 \times \cdots \times I_N \times M}$, $G \in R^{I_1 \times \cdots \times I_N \times M}$, and $A_n \in R^{I_n \times J_n n = 1, ..., N}$. $\Pi \in R^{M \times M}$ also is a graph Laplacian matrix. If the row vectors of $X (N)$ are regarded as data, then the row vectors of $G (N)$ are the compact or dimension-reduced representation of data.

Orthographical constraint Tucker decomposition is another research topic of Tucker decomposition. For example, MMF [24] learned the low-rank factorization with manifold constraints, which is described as

$$\min_{U \geq 0, Y \geq 0} \| \mathcal{X} - U Y \|_F^2 + \lambda tr (Y \Pi Y^T), \quad (13)$$

here, the requirement $U^T U = I_d$ represents low-rank approximation. The MMF algorithm incorporates manifold regularization into matrix factorization. The new regularization model is better than graph regularization non-negative matrix factorization, and has global optimal solutions and closed-form solutions. It is argued in [24] that MMF achieves better results than GNMF in some applications.

Zhou et al. [27] introduced non-negative Tucker decomposition methods to approximate the low-dimensional tensor with effective projection matrices, such as

$$\min_{G, A_n \geq 0} \| \mathcal{T} - G X_1 A_1 \cdots X_{N-1} A_{N-1} X_N A_N \|_F^2. \quad (14)$$
This objective function for the last mode matrix $A_N$ is a function on Riemannian manifold. At present, most optimizations for objective functions defined on Riemannian manifold only involve the gradients of objective functions (first-order Riemannian geometry). HTD expands the objective function from first-order Riemannian geometry (gradients) up to second-order Riemannian geometry (Hessians).

LRR-HTD [28], as a famous TDR approach, is developed from NTD

$$\min_{Q, A_1, \ldots, A_N} \|X - QX_{1}A_1 \cdots X_{N-1}A_{N-1}X_NA_N\|^2 + \lambda \text{rank}(A_N),$$

(15)

where $X \in \mathbb{R}^{L_1 \times \cdots \times L_{N-1} \times M}$, $A_n \in \mathbb{R}^{L_n \times J_n}$, $n = 1, \ldots, N - 1$ and $A_N \in \mathbb{R}^{M \times J_M}$. LRR-HTD combined various constraints in different modes to enhance the robustness of the proposed algorithm. Specifically, due to the noise and redundancy in the original tensor, LRR-HTD found a set of orthogonal factor matrices for all modes to map the tensor to a low-dimensional subspace.

Zhang et al. [26] introduced the local homeomorphism characters of manifold data and global subspace mapping to TDR, named LHGPD,

$$\min_{W} \|W W^T - I\|_F = \min_{W} \|W^T (X^{(N)})^T X^{(N)} W^T - I\|_F,$$

(16)

where $W$ denotes the projection matrix of subspace. Actually, LHGPD and MRNTD introduced the idea of classical manifold learning algorithms local tangent space alignment (LTSA) [51] and Laplacian eigenmaps (LE) [52] to TDR, respectively. Different from these methods, our proposed algorithm introduce the statistical information of the original data set to learn the low-dimensional representation of tensor for improving the performance of TDR.

4 | THE PROPOSED ALGORITHM

In this section, we propose a novel TDR method that explores the statistical information of tensor data set by HSIC and learns an explicit expression between the dimensionality-reduced tensors and the higher-dimensional tensors via mode product.

4.1 | The framework of HSIC-TDR

HSIC cannot to be directly applied to tensor DR. Therefore, we first derive the mathematical processes of HSIC for tensor data and establish a framework of TDR based on HSIC, named HSIC-TDR.

We denote by $X \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_{N-1} \times M}$ the $N - 1$-dimensional multi-dimensional data sets to be used. And $M$ denotes the number of tensor data included in multi-dimensional data set. Then, the $N$ mode of $X$ can be described as

$$X^{(N)} = \begin{bmatrix} X^{(N)}_{1, \text{Row}} \\ \vdots \\ X^{(N)}_{M, \text{Row}} \end{bmatrix},$$

(17)

The aim of DR for tensor $X$ is to find a low-dimensional tensor $G \in \mathbb{R}^{L_1 \times \cdots \times L_{N-1} \times M}$ where $J_n < L_n$, $n = 1, \ldots, N - 1$. The $N$ mode of $G$ is

$$G^{(N)} = \begin{bmatrix} g_1^{(N)}_{1, \text{Row}} \\ \vdots \\ g_{J_n}^{(N)}_{1, \text{Row}} \\ \vdots \\ g_1^{(N)}_{M, \text{Row}} \\ \vdots \\ g_{J_n}^{(N)}_{M, \text{Row}} \end{bmatrix} \in \mathbb{R}^{M \times (J_1 \cdots J_{N-1})}.$$

Note that the number of $X$ and $G$ is the same. Except it, each dimension of $G$ is smaller than the corresponding dimension of $X$. Therefore, $G$ is the low-dimensional representation of $X$.

With $X$ and $G$, we can establish an objective function for TDR by the criterion of HSIC maximization

$$\max_{G} \text{HSIC}(X, G) = \max_{G^{(N)}} \text{HSIC}(X^{(N)}, G^{(N)})$$

$$= \max_{G^{(N)}} \text{tr} (K_G C_M K_X C_M).$$

(18)

Here, $C_M = I_M - \frac{1}{M} \Gamma_M \Gamma_M^T$ is a centralizing matrix, $k_G$ and $k_X$ are kernel functions,

$$K_G = \begin{bmatrix} k_G (G^{(N)}_{1, \text{Row}}, G^{(N)}_{1, \text{Row}}) & \cdots & k_G (G^{(N)}_{1, \text{Row}}, G^{(N)}_{M, \text{Row}}) \\ \vdots & \ddots & \vdots \\ k_G (G^{(N)}_{M, \text{Row}}, G^{(N)}_{1, \text{Row}}) & \cdots & k_G (G^{(N)}_{M, \text{Row}}, G^{(N)}_{M, \text{Row}}) \end{bmatrix},$$

(19)

$$K_X = \begin{bmatrix} k_X (X^{(N)}_{1, \text{Row}}, X^{(N)}_{1, \text{Row}}) & \cdots & k_X (X^{(N)}_{1, \text{Row}}, X^{(N)}_{M, \text{Row}}) \\ \vdots & \ddots & \vdots \\ k_X (X^{(N)}_{M, \text{Row}}, X^{(N)}_{1, \text{Row}}) & \cdots & k_X (X^{(N)}_{M, \text{Row}}, X^{(N)}_{M, \text{Row}}) \end{bmatrix}.$$
positive definiteness, it can be used as kernel function. Therefore, the selection of \( k_G \) is an open problem. The learning problem of HSIC-TDR is an optimization problem of the \( k_G \) with \( k_G \). If \( k_G \) is non-linear, the difficulty of the optimization process of (18) can be increased. Therefore, we choose a linear kernel function for \( G \)

\[
k_G(x,y) = \kappa^T y + \kappa \delta(x,y),
\]

where \( \kappa > 0 \), \( \delta(x,y) = \begin{cases} 1, & x = y \\ 0, & \text{otherwise} \end{cases} \). \( \delta(x,y) \) is added to guarantee the positive determinate of \( k_G(x,y) \). Then, the kernel matrix \( K_G \) is

\[
K_G = \begin{bmatrix}
    k_G(G_{1Row}^{(N)} G_{1Row}^{(N)}) & \cdots & k_G(G_{1Row}^{(N)} G_{MRow}^{(N)}) \\
    \vdots & \ddots & \vdots \\
    k_G(G_{MRow}^{(N)} G_{1Row}^{(N)}) & \cdots & k_G(G_{MRow}^{(N)} G_{MRow}^{(N)})
\end{bmatrix}
\]

\[
= \begin{bmatrix}
    G_{1Row}^{(N)} G_{1Row}^{(N)}^T & \cdots & G_{1Row}^{(N)} G_{MRow}^{(N)}^T \\
    \vdots & \ddots & \vdots \\
    G_{MRow}^{(N)} G_{1Row}^{(N)}^T & \cdots & G_{MRow}^{(N)} G_{MRow}^{(N)}^T
\end{bmatrix} + \kappa I_{J_1\ldots J_{N-1}}.
\]

Substituting Equation (22) into the objective function of HSIC-TDR, then Equation (18) can be transformed into

\[
tr(K_G C_M K_X C_M) = tr(G^{(N)} G^{(N)}^T C_M K_X C_M)
\]

\[
+ \kappa tr(C_M K_X C_M)
\]

\[
= tr\left( (G^{(N)} G^{(N)}^T C_M K_X C_M) \right)
\]

\[
+ \kappa tr(C_M K_X C_M),
\]

(23)

\( tr(C_M K_X C_M) \) has nothing to do with the optimization objective (23), thus the objective function of HSIC-TDR can be simplified as:

\[
\max_{G^{(N)}} tr\left( (G^{(N)} G^{(N)}^T C_M K_X C_M) \right).
\]

(24)

In the objective function of HSIC-TDR (23), \( k_X \) is the kernel function of the high-dimensional tensor data space \( \Omega_X \), that is, \( \Omega_X \subset \mathbb{R}^{l_1 \times \ldots \times l_{N-1}} \), \( k_X : \Omega_X \times \Omega_X \to \mathbb{R} \). And \( k_X \) is symmetrical positive definiteness. Different with \( k_Y \), the choice of \( k_X \) is open in our method, as long as meets the condition of symmetry positive definite. Therefore, the framework of HSIC-TDR can generate different algorithms by selecting different kernel functions or learning different kernel functions for different application occasions.

### 4.2 MPHSIC-TDR

In general, the task of TDR can be expressed as

\[
\mathcal{X} \in \mathbb{R}^{l_1 \times \ldots \times l_{N-1} \times M} \Rightarrow G \in \mathbb{R}^{l_1 \times \ldots \times l_{N-1} \times M}.
\]

Here, \( f_n \ll l_m, n = 1, \ldots, N-1 \).

According to tensor algebra, the mode product of tensor with a matrix can change the dimension of the tensor in certain order. In the section, we first propose an algorithm for TDR based on mode product (MP-TDR):

\[
G = A \mathcal{X}_1 A_1^T \ldots \mathcal{X}_{N-1} A_{N-1}^T = \mathcal{X} N \prod_{i=1}^{N-1} \mathcal{X}_i A_i^T,
\]

where \( A_i \subset \mathbb{R}^{l_i \times l_i}, n = 1, \ldots, N-1 \). According to the definition of the mode product, \( G \) satisfies the requirements of DR for the high-dimensional tensor \( \mathcal{X} \).

MP-TDR gives an explicit expression between the dimensionality-reduced tensor and the high-dimensional tensor \( G = \mathcal{X} \prod_{i=1}^{N-1} \mathcal{X}_i A_i^T \). Note that the determination of the mode matrix \( A_i \) is open. Therefore, different strategies under MP-TDR maybe lead to different algorithms.

This paper employs the HSIC maximization criterion to learn the matrix \( A_1, \ldots, A_{N-1} \) for TDR, that is,

\[
\max_{G^{(N)}} HSIC(\mathcal{X}, G) = \max_{A_1, \ldots, A_{N-1}} HSIC\left( \mathcal{X}, \mathcal{X} \prod_{i=1}^{N-1} \mathcal{X}_i A_i^T \right).
\]

Furthermore, according to the theorem of tensor algebra, we have

\[
G = \mathcal{X} A_1 A_1^T \ldots \mathcal{X}_{N-1} A_{N-1}^T \Rightarrow G^{(N)} = \mathcal{X}^{(N)} (A_1 \otimes \cdots \otimes A_{N-1}) = \mathcal{X}^{(N)} W,
\]

(27)

where \( W = A_1 \otimes \cdots \otimes A_{N-1} \otimes A_{N-1} \in \mathbb{R}^{(l_1 \times \ldots \times l_{N-1}) \times (l_1 \times \ldots \times l_{N-1})} \).

Therefore, the objective function of MPHSIC-TDR algorithm based on the model product and HSIC can be expressed as

\[
\max_{A_1, \ldots, A_{N-1}} HSIC\left( \mathcal{X}, \mathcal{X} \prod_{i=1}^{N-1} \mathcal{X}_i A_i^T \right)
\]

\[
= \max_{G^{(N)}} tr\left( (G^{(N)} G^{(N)}^T C_M K_X C_M) \right)
\]

\[
= \max_{W} tr\left( W^T (\mathcal{X}^{(N)} \otimes \mathcal{X}^{(N)} W) \right).
\]

(28)
Here, $\Pi = (\mathcal{X}^{(N)})^T C_M K \bar{C}_M \mathcal{X}^{(N)} \in R^{(L_1 \cdots L_{N-1}) \times (L_1 \cdots L_{N-1})}$ is a symmetric semi-positive definite matrix. MPHSIC-TDR not only realizes the DR of tensor data, but also gives the explicit expression between the high-dimensional tensor $\mathcal{X}$ and the DR tensor $\mathcal{G}$, i.e. $\mathcal{G}^{(N)} = \mathcal{X}^{(N)} W'$. For the new set of tensor data $\mathcal{X}' \in R^{L_1 \cdots L_{N-1} \times H}$ and the data distribution of $\mathcal{X}'$ and $\mathcal{X}$ is the same, then $\mathcal{X}'$ can be reduced by using the formula $\mathcal{G}(N) = \mathcal{X}'(N) W'$, namely,

$$\mathcal{Z}(N) = \mathcal{X}'(N) W', \quad (29)$$

where $\mathcal{Z} \in R^{(H \times \cdots \times N/1) \times H}$ is a dimensionality-reduced tensor of $\mathcal{X}'$, $H \leq 1$ is the number of data $\mathcal{Y}$ that contains.

Furthermore, we solve the learn problem of MPHSIC-TDR by adding constraints $A_i^T \Lambda_i A_i = R_i$, $i = 1, \ldots, N - 1$. Therefore, there is

$$W'^T W' = (A_1 \otimes \cdots A_{N-1})^T (A_1 \otimes \cdots A_{N-1})$$

$$= (A_i^T A_1 \otimes \cdots \otimes A_{N-1})$$

$$= I_{j_1} \cdots I_{j_{N-1}}.$$

At the same time, the learning problem of MPHSIC-TDR becomes the problem of Rayleigh quotient. The column vectors of the matrix $W'$ are the $j_1 \cdots j_{N-1}$ basis of the symmetric semi-positive definite matrix $\Pi$, i.e. the orthogonal eigenvectors corresponding to the largest eigenvalues $j_1 \cdots j_{N-1}$ of $\Pi$.

### 4.3 Complexity analysis

In this section, we analyse the complexity of our algorithm. The computational complexity can be started from four aspects: the calculation of $C_M$, the kernel matrix $K_{\bar{X}}$ of the original tensor, $C_M K_{\bar{X}} C_M$ and the dimensionality-reduced matrix $G^{(N)}$. $C_M$ can be viewed as the product of a column vector and a row vector, so the complexity of $C_M$ is $O(M^2)$. Kernel function is the mapping of two vectors to a real number, where the inner product of two vectors needs to be constructed, therefore the complexity of kernel function is $O(N^2)$. Since the kernel matrix has $M^2$ elements, all the above complexity is $O(M^2 \prod_{i=1}^{N-1} L_i)$. The complexity of eigenvalue decomposition of $C_M K_{\bar{X}} C_M$ is $O((\prod_{i=1}^{N-1} L_i)^2)$. The last step is to obtain the matrix $G^{(N)}$ after DR, which can be obtained by Equation (27). Therefore, the complexity of multiplying between two matrices needs to be calculated that is $O(M^2 \prod_{i=1}^{N-1} L_i^2 \prod_{i=1}^{N-1} j_i)$. Finally, the total computational complexity of the algorithm is $O(M^2 + M^2 \prod_{i=1}^{N-1} L_i + (\prod_{i=1}^{N-1} L_i)^2 + M(\prod_{i=1}^{N-1} L_i)^2 \prod_{i=1}^{N-1} j_i)$.

### 5 | EXPERIMENTAL RESULTS

In this section, we verify the effectiveness of the proposed algorithm by conducting the clustering and classification experiments on six data sets (COIL20, ETH80, ORL, USPS, Faces94 males and Olivetti), and compare it with GLTD [25], GNMF [29], MMF [24], LRR-HTD [28], MRNTD [50], LHGPD [26] and NTD [27] (see Table 1).

#### 5.1 Data sets

COIL20 [53] contains 20 objects taken from different angles, seen from Figure 1, and 72 images per object. Each image is uniformly processed as the size of 32*32. Furthermore, we have transformed the entire data set as a tensor of 32*32*1440.

ETH80 (see Figure 2) [54] is a multi-visual image database, which contains 8 categories and 10 objects and 41 different visual images with the size of 128*128 in each object. In the experiments, the size is reset to 32*32 and the all images are expressed as a tensor object of 32*32*3280.

ORL face data set [55] (see Figure 3) consists of 400 facial objects with 40 images under different poses and expressions. The size of each image is 32*32. We restore the data set to the form of 32*32*400 for the experiments.

USPS [56] (see Figure 4), another famous handwritten digit data set, consists of a total of 2000 handwritten digital images. The entire data set is converted to a tensor form of 16*16*2000.

| Algorithms | Description |
|------------|-------------|
| NTD        | Non-negative Tucker decomposition for TDR |
| GNMF       | Introducing data graph to non-negative matrix decomposition |
| MMF        | Introducing low-rank regularization to matrix factorization |
| GLTD       | Introducing graph-Laplacian to Tucker tensor decomposition |
| MRNTD      | Manifold canonical non-negative Tucker decomposition |
| LRR-HTD    | Low-rank regularized heterogeneous tensor decomposition |
| LHGPD      | Local homeomorphism subspace mapping for TDR |
| MPHSIC-TDR | HSIC maximization subspace mapping for TDR |

#### Table 1 Description of compared algorithms
FIGURE 1  Partial samples of COIL20 data set

FIGURE 2  Partial samples of ETH80 data set
Faces94 males data set [57] (see Figure 5) contains 153 subjects (20 female subjects, 113 male subjects and 20 male staff subjects). And each subject has 20 images with different facial expressions. All the images are resized to 50 * 45 in our experiment.

Olivetti Faces [58], a relatively small face database of New York university, consists of 400 pictures of 40 people. Each person has 10 face images. And the size of each picture is 64*64 (see Figure 6).

5.2 Parameter selection

In the experiments, we choose the suitable kernel function from multiple kernel functions for different data sets. The kernel function set contains six typical kernel functions: the Gaussian kernel, the wavelet kernel, the GTS (Generalized T-Student), the hyperbolic tangent kernel (Tanh), the sigmoid kernel and linear kernel.
In classification experiments, the KNN classifier is implemented to test the classification accuracy of all the compared algorithm and its nearest neighbour number is set to 10. In clustering experiments, the $K$-means is utilized to collect clustering information of the proposed algorithm. The reduced dimensions of the eight data sets (COIL20, ETH80, ORL, MNIST, USPS, CMU PIE, Faces94 males and Olivetti) are experimentally set to be $8 \times 8$, $8 \times 8$, $6 \times 6$, $8 \times 8$, $7 \times 8$, $8 \times 8$ and $10 \times 8$, respectively.

### 5.3 Clustering experimental results

In this section, we aim to demonstrate the TDR capability of our proposed algorithm by performing clustering experiments on the reduced tensor data. Clustering accuracy (AC) and normal mutual information (NMI) are used to evaluate the performances of the compared algorithms.

AC is defined as

\[
Clustering \ Accuracy = \frac{1}{M} \sum_{j=1}^{M} \delta(l_j^*, l_j),
\]

where $l_j^*$, $l_j$ denote true label and clustering label, respectively. When $l_j^* = l_j$, $\delta(\cdot, \cdot) = 1$, otherwise, $\delta(\cdot, \cdot) = 0$.

NMI is defined as

\[
NMI(l^*, l) = \frac{MI(l^*, l)}{\max(H(l^*), H(l))},
\]

where $H(l^*)$ and $H(l)$ denote the entropy of $l^*$ and $l$, respectively. $MI$ denotes the mutual information of $l^*$ and $l$.

The clustering method adopts the distance-based $K$-means for data clustering. It means that the closer the distance between two objects is, the greater the similarity will be. Since the $K$-means algorithm is sensitive to the selection of initial data, we repeat the clustering process 10 times in each compared algorithm experiment, and take the average results as the finally experiment results. On each data set, we randomly select $D$ samples from $C$ classes as the training set for clustering experiments. Then 10 training sets are selected, and the average of the clustering results is calculated at the final.

For the ETH80 data set, we randomly select 41 samples from each class as the training set for clustering experiments. In the following, we first conduct the experiments of selecting the best kernel function for the proposed algorithm based
on the AC and NMI, respectively (see Table 2). It shows that under different clustering criterion, the best kernel maybe different. With the decrease of the data size and number of classes, the CA and clustering NMI increase significantly. We randomly select 20 samples from each class of Faces94 data set as the training set for clustering experiments. The experiment results of selecting the best kernel function for MPHSIC-TDR are shown in Table 3. It can be seen that with different clustering criterion, the best kernel also is different. It means that different data sets and different evaluation criterion need different kernel function. With the decrease of the data size and number of classes, the CA and clustering NMI on Faces94 also increase significantly. The reason may be that single kernel function performs well in less categories. Inspired by it, the research of multiple kernels for HSIC will be one of our further work.

### TABLE 2
The clustering results on ETH80 for the best kernel function selection of MPHSIC-TDR

| AC | D | C | Gaussian | Wavelet | GTS | Tanh | Sigmoid | Best kernel |
|----|---|---|----------|---------|-----|------|---------|-------------|
| 328 | 8 | 0.6413 | 0.6271 | 0.6117 | 0.6388 | 0.6334 | **0.6413** |
| 246 | 6 | 0.7125 | 0.7505 | 0.7510 | 0.7337 | 0.7404 | **0.7505** |
| 164 | 4 | 0.8893 | 0.8836 | 0.8893 | 0.6120 | 0.8871 | **0.8893** |
| 82  | 2 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | **1.0000** |

| NMI | D | C | Gaussian | Wavelet | GTS | Tanh | Sigmoid | Best kernel |
|-----|---|---|----------|---------|-----|------|---------|-------------|
| 328 | 8 | 0.6355 | 0.6242 | 0.6405 | 0.6254 | 0.6335 | **0.6405** |
| 246 | 6 | 0.6179 | 0.6701 | 0.6799 | 0.6373 | 0.6799 | **0.6799** |
| 164 | 4 | 0.7628 | 0.7788 | 0.7811 | 0.4784 | 0.7726 | **0.7788** |
| 82  | 2 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | **1.0000** |

### TABLE 3
The clustering results on Faces94 for the best kernel function selection of MPHSIC-TDR

| AC | D | C | Gaussian | Wavelet | GTS | Tanh | Sigmoid | Best kernel |
|----|---|---|----------|---------|-----|------|---------|-------------|
| 600 | 30| 0.8290 | 0.8077 | 0.8120 | 0.4530 | 0.8005 | **0.8290** |
| 400 | 20| 0.8377 | 0.8624 | 0.8511 | 0.3121 | 0.8347 | **0.8624** |
| 200 | 10| 0.9010 | 0.8918 | 0.9040 | 0.5518 | 0.9200 | **0.9200** |

| NMI | D | C | Gaussian | Wavelet | GTS | Tanh | Sigmoid | Best kernel |
|-----|---|---|----------|---------|-----|------|---------|-------------|
| 600 | 30| 0.8290 | 0.8046 | 0.8210 | 0.3842 | 0.8115 | **0.8290** |
| 400 | 20| 0.8115 | 0.8228 | 0.8625 | 0.2984 | 0.8453 | **0.8625** |
| 200 | 10| 0.9229 | 0.9009 | 0.8743 | 0.4978 | 0.8813 | **0.9229** |

Note: The clustering accuracy and NMI performances of MPHSIC-TDR with different kernel functions versus different number of categories (C) on ETH80 are shown here. (D is the number of data samples).

Note: The clustering accuracy and NMI performances of MPHSIC-TDR with different kernel functions versus different number of categories (C) on Faces94 are shown here. (D is the number of data samples).
FIGURE 7  The clustering experimental results on ETH80, and Faces94 data sets. The clustering accuracy and NMI performances of eight compared algorithms versus different number of categories (C) on ETH80, and Faces94 male are shown here. The first row shows the AC and NMI on ETH80, and the second row shows the AC and NMI on Faces94 data set. MPHSIC-TDR is the proposed algorithm.

Seen from Figures 7(a) and (b), most of algorithms can obtain a good performance when the number of categories is small. Although the clustering results of each algorithm ETH80 have a little different under different computed criteria, their AC and NMI performances are basically consistent with the increasing of the number of categories. In other words, GNMF has the worst performances on ETH80, and GLTD follows it. In most cases, LHGPD and MRNTD have a little better than GLTD. The performance of MMF is better than LHGPD and MRNTD. But MMF is not stable, such as when $C = 6$, it works worse than LHGPD and MRNTD. LRR-HTD and our MPHSIC-TDR have the best performance when the number of categories is small. But with increasing of the number of categories, the performance of LRR-HTD has deteriorated a lot. Seen from Figures 7(c) and (d), MRNTD and LHGPD almost have the worst performances on Faces94 male data set, and show poor robustness. MMF and NTD have different performance under different clustering criteria. GLTD and GNMF work better on the Faces94 data set. On the whole, our MPHSIC-TDR also has the best performance. The reason is that we choose the best kernel function, i.e. the most suitable kernel, for each data set. Therefore, the corresponding RKHS generated from the best kernel function contains unique information of data set. Then MPHSIC-TDR can explore the global probability distribution information for TDR.

In order to show more direct observation, the average clustering results of eight compared algorithms on ETH80 and Faces94 male data sets are shown in Figure 8 for visualization. It can been seen that the experimental results of our proposed algorithm is higher than other compared algorithms. The clustering experimental results verify the effectiveness of the proposed algorithm.

5.4  Classification experimental results

In the classification experiments, we adopt the KNN classifier with $k = 10$ on reduced-dimensional data sets, and use the cross-validation method for computing the accuracy. For each data set, we first conduct the experiments of selecting the best kernel function for the proposed algorithm based on classification accuracy. We randomly select $D$ samples from $C$ classes as the sample set for classification experiments. For example, we randomly select 72, 41, 10, 200 and 20 samples from each class of the six data sets (COIL20, ETH80, ORL, USPS, Faces94 males and Olivetti) as the sample set, respectively. The compared algorithms contains GNMF, MMF, LRR-HTD, MRNTD, LHGPD, NTD and MPHSIC-TDR, and MPHSIC-TDR is the proposed algorithm. In each experiment, we randomly select 80% of the reduced-dimensional data as the training set and the remaining 20% as the testing set for classification.
The average clustering results of eight compared algorithms on ETH80 and Faces94 male data sets. The average clustering accuracy and average NMI performances of eight compared algorithms on ETH80, and Faces94 male are shown here. MPHSIC-TDR is the proposed algorithm.

| $D$  | $C$  | Gaussian | Wavelet | GTS   | Tanh  | Sigmoid | Best kernel |
|------|------|----------|---------|-------|-------|---------|-------------|
| 1440 | 20   | 0.9740   | 0.9635  | 0.9740 | 0.8177 | 0.9722  | 0.9740      |
| 1296 | 18   | 0.9768   | 0.9711  | 0.9942 | 0.8228 | 0.9865  | 0.9942      |
| 1152 | 16   | 0.9804   | 0.9696  | 0.9848 | 0.6398 | 0.9848  | 0.9848      |
| 1008 | 14   | 0.9726   | 0.9827  | 0.9703 | 0.3374 | 0.9851  | 0.9851      |
| 864  | 12   | 0.9624   | 0.9739  | 0.9856 | 0.6119 | 0.9854  | 0.9856      |
| 720  | 10   | 0.9549   | 0.9688  | 0.9722 | 0.8229 | 0.9792  | 0.9792      |
| 576  | 8    | 0.9609   | 0.9608  | 0.9651 | 0.8134 | 0.9738  | 0.9738      |
| 432  | 6    | 0.9884   | 0.9651  | 0.9884 | 0.8839 | 0.9767  | 0.9884      |
| 288  | 4    | 1.0000   | 0.9569  | 1.0000 | 0.9477 | 0.9738  | 1.0000      |
| 144  | 2    | 1.0000   | 1.0000  | 1.0000 | 0.9476 | 1.0000  | 1.0000      |

Note. The classification results on COIL20 for the best kernel function selection of MPHSIC-TDR. The classification accuracy of MPHSIC-TDR with different kernel functions versus different number of categories ($C$) on COIL20 are shown here. ($D$ is the number of data samples)

Each experiment tests 20 times and the average classification accuracy as the final result.

Table 4 lists the classification results versus different number of categories ($C$) for the best kernel function selection of MPHSIC-TDR in COIL20. From Table 4, we see that the best kernel is different under different number of categories. The classification results of seven compared methods for COIL20 data set are shown in Figure 9(a). It can be seen that all methods have a good performance when the number of categories is small. GNMF get the worst results, and MMF follows it. The reason may be that they are sensitive to the change of the number of neighbours. The performances of LRR-HTD are still not stable as the number of categories increase. NTD and MRNTD work better than the above three algorithms. LHGPD shows better effectiveness in the task of classification. MPHSIC-TDR has the best performance in the task of classification, which can achieve the separability of distribution information of different categories in COIL20.

Table 5 exhibits the classification results versus different number of categories ($C$) for the best kernel function selection of MPHSIC-TDR on ETH80. Figure 9(b) shows the classification performances of seven algorithms on ETH80. First, we observed that overall, the compared methods have a good classification results in ETH80. But when the number of categories starts to grow, the classification accuracy of them are decrease. However, MPHSIC-TDR still outperforms the other methods. It proves that the proposed algorithm can effectively explore the global statistical information of tensor data set with learning the best kernel function.

Table 6 lists the classification results versus different number of categories ($C$) for the best kernel function selection of MPHSIC-TDR on ORL. Figure 9(c) shows the classification performances of seven compared algorithms on ORL data set. From this figure, we see that NTD almost has the worst performance, and MRNTD follows it. This may be because that there are too few samples in ORL data to obtain an effective NTD. LHGPD perform better in the task of classification on ORL. Overall, our proposed MPHSIC-TDR has got the best classification results in most case.

Table 7 shows the classification results versus different number of categories ($C$) for the best kernel function selection of MPHSIC-TDR on USPS data set. Figure 9(d) shows the classification performances versus different number of categories in seven algorithms on USPS. Similar to the classification results on ORL, the performance of NTD and MRNTD on the USPS data set are poor. When the number of class is less than 8, the performance of GNMF and LRR-HTD are similar. The performance of MMF is close to LHGPD. And the classification performances of MPHSIC-TDR on this data set still are satisfactory.

Table 8 lists the classification results versus different number of categories ($C$) for the best kernel function selection of MPHSIC-TDR in Faces94 males data set. Figure 9(e) and
The classification experimental results on (a) COIL20, (b) ETH80, (c) ORL, (d) USPS, (e) Faces94 and (f) Olivetti data sets. The classification of eight compared algorithms versus different number of categories ($C$) on COIL20, ETH80, ORL, USPS, Faces94 and Olivetti are shown here. MPHSIC-TDR is the proposed algorithm.

Table 9 show the classification performances of all compared algorithms on Faces94 males. From them, we can observed that GNMF and MPHSIC-TDR (our method) have the best performance. The reason maybe that matrix-based method may have a better performance than some high-order-tensor-based method on some data set. And the MPHSIC-TDR has a stable and good performance by preserving the global statistical information.
### TABLE 9  The classification experimental results on Faces94 data set

| D  | C   | LRR  | MR  | MNMF | MMF  | HTD  | NTD  | LHGPD | NTD  | TDR  |
|----|-----|------|-----|------|------|------|------|-------|------|------|
| 600| 30  | 1.0000 | 0.9967 | 0.9883 | 0.7583 | 0.9843 | 0.9700 | 1.0000 |
| 400| 20  | 1.0000 | 0.9875 | 0.8375 | 0.9753 | 0.9550 | 1.0000 |
| 200| 10  | 1.0000 | 0.9870 | 0.7050 | 0.9669 | 0.9200 | 1.0000 |

Note: The classification of eight compared algorithm versus different number of categories (C) on Faces94 are shown here. MPHSIC-TDR is the proposed algorithm.

### TABLE 10  Classification accuracy of MPHSIC-TDR with different kernel on Olivetti data set (D data size, C number of classes)

| D   | C   | Gaussian | Wavelet | GTS | Tanh | Sigmoid | Best kernel |
|-----|-----|----------|---------|-----|------|---------|-------------|
| 400 | 40  | 0.7000   | 0.6813  | 0.7375 | 0.3875 | 0.7063  | 0.7375      |
| 300 | 30  | 0.7083   | 0.7250  | 0.6917 | 0.3167 | 0.7833  | 0.7833      |
| 200 | 20  | 0.7125   | 0.7375  | 0.6500 | 0.2875 | 0.6750  | 0.7375      |
| 100 | 10  | 0.7500   | 0.7750  | 0.6750 | 0.4000 | 0.7250  | 0.7750      |

Note: The classification results on Olivetti for the best kernel function selection of MPHSIC-TDR. The classification accuracy of MPHSIC-TDR with different kernel functions versus different number of categories (C) on Olivetti are shown here. (D is the number of data samples)

Table 10 lists the classification results versus different number of categories (C) for the best kernel function selection of MPHSIC-TDR on Olivetti data set. Figure 9(f) shows the classification performances of all compared methods on Olivetti. Seen from Figure 9(f), the performance of MRNTD is worst, and MMF and LRR-HTD have a higher classification results than it. And obviously, our proposed MPHSIC-TDR has got the best classification. Moreover, the effectiveness obtained by the proposed MPHSIC-TDR method on Olivetti data set is significant compared with the six state-of-the-art algorithms.

To observe the classification performances of all methods numerically, Table 11 summarizes the average classification results of each algorithm on six data sets. We also provide the standard deviation of the average classification results in the last line of Table 11 to show the robustness of our algorithm. Obviously, the proposed MRHSIC-TDR performs better than other algorithms, which confirms its effectiveness in data-classification task. According to the experiment tables and figures, it is verified that HSIC criterion can be applied to the algorithm of data DR, which enriches the algorithm library of data DR to some certain extent. The validity and reliability of the proposed algorithm can be verified by combining the experimental results of clustering and classification.

### 6  CONCLUSION

This paper proposes a novel approach based on mode product and HSIC. HSIC criterion measures the statistical correlation between two data sets. However, instead of directly measuring the statistical correlation between two data sets, HSIC first transforms them to two RKHSs, and then uses the HS operator of the two RKHSs to measure the statistical information of data sets. The RKHS can be generated from different kernel function according to different application situation, increasing the applicability and flexibility of HSIC. In our work, we first establish a novel framework for TDR (named HSIC-TDR), which captures statistical information of tensor data set. According to tensor algebra, the dimensionality of tensor can be changed by the mode product of the tensor and a matrix. Based on it, MP-TDR is generated for learning an explicit expression between the high-dimensional tensor and its low-dimensional representation. In MP-TDR, the determination of the mode product matrix is open. Furthermore, we propose a novel algorithm (MPHSIC-TDR), which has the merits of both HSIC and mode product. MPHSIC-TDR not only explores the global statical information of tensor data set, but also learns an explicit mapping for out-of-samples. Finally, we conduct the clustering and classification experiments on six real-world data sets with some state-of-the-art algorithms to demonstrate the effectiveness of the proposed method.

### ACKNOWLEDGEMENTS

This work is supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant.

### TABLE 11  The average classification results of eight compared algorithms on COIL20, ETH80, ORL, USPS, Olivetti and Faces94 male data sets

| Data set | GNMF | MMF | HTD | NTD | LHGPD | NTD | MPHSIC-TDR |
|----------|------|-----|-----|-----|-------|-----|------------|
| COIL20   | 0.8740 | 0.9075 | 0.9055 | 0.9424 | 0.9675 | 0.9339 | 0.9865 |
| ETH80    | 0.7471 | 0.8865 | 0.9022 | 0.8487 | 0.9076 | 0.8489 | 0.9131 |
| ORL      | 0.4156 | 0.5596 | 0.5305 | 0.3131 | 0.7495 | 0.2856 | 0.7891 |
| USPS     | 0.9203 | 0.9438 | 0.8946 | 0.7827 | 0.9404 | 0.7973 | 0.9624 |
| Faces94 male | 1.0000 | 0.9989 | 0.9876 | 0.7669 | 0.9755 | 0.9483 | 1.0000 |
| Olivetti | 0.4490 | 0.5731 | 0.6169 | 0.2975 | 0.6596 | 0.4552 | 0.7583 |

Standard deviation | 0.2481 | 0.1938 | 0.1853 | 0.2806 | 0.1309 | 0.2753 | 0.1038 |

Note: The average classification accuracy performances of eight compared algorithms on COIL20, ETH80, ORL, USPS, Olivetti and Faces94 male are shown here. MPHSIC-TDR is the proposed algorithm.
2019A1515111143 and by the Natural Science Foundation of China under Grant 62006042.

ORCID
Zhengming Ma https://orcid.org/0000-0001-6553-1070

REFERENCES
1. Lu, W., et al.: Web multimedia object clustering via information fusion. In: IEEE International Conference on Document Analysis and Recognition, pp. 319–323 (2011)
2. Denli, H., Subrahmanya, N.: Multi-scale graphical models for spatio-temporal processes. In: Advances in Neural Information Processing Systems, pp. 316–324 (2014)
3. Zeng, E., et al.: Clustering genes using heterogeneous data sources. In: Computational Knowledge Discovery for Bioinformatics Research, pp. 67–83. IGI Global (2012)
4. van der Maaten, L., Postma, E., van den Herik, J.: Dimensionality reduction: A comparative. J. Mach. Learn. Res. 10(13), 66–71 (2009)
5. Bengua, J.A., et al.: Efficient tensor completion for color image and video recovery: Low-rank tensor train. IEEE Trans. Image Process. 26(5), 2466–2479 (2017)
6. Yang, J., et al.: Two-dimensional PCA: A new approach to appearance-based face representation and recognition. IEEE Trans. Pattern Anal. Mach. Intell. 26(1), 131–1370 (2004)
7. Li, N., Li, B.: Tensor completion for on-board compression of hyperspectral images. In: IEEE International Conference on Image Processing, pp. 517–520 (2010)
8. Zhang, S., Ma, Z.M., Gan, W.: Dimensionality reduction for tensor data based on local decision margin maximization. IEEE Trans. Image Process. 30, 234–248 (2020)
9. Karatzoglou, A., et al.: Multiverse recommendation: n-Dimensional tensor factorization for context-aware collaborative filtering. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 79–86 (2010)
10. Sun, WW, et al.: Provable sparse tensor decomposition. J. R. Stat. Soc.: Ser. B. Stat. Method. 79(3), 899–916 (2017)
11. Tang, M., et al.: A joint tensor completion and prediction scheme for multi-dimensional spectrum map construction. IEEE Access 4, 8044–8052 (2016)
12. Zhang, Y., et al.: Massive 3D seismic data compression and inversion with hierarchical Tucker. In: SEG International Exposition and Annual Meeting, Society of Exploration Geophysicists (2017)
13. Liu, J., et al.: Tensor completion for estimating missing values in visual data. IEEE Trans. Pattern Anal. Mach. Intell. 35(1), 208–220 (2012)
14. Shi, G.Y., Huang, H., Wang, L.H.: Unsupervised dimensionality reduction for hyperspectral imagery via local geometric structure feature learning. IEEE Geosci. Remote Sens. Lett. 17(8), 1425–1429 (2020)
15. Luo, F.L., et al.: Feature learning using spatial-spectral hypergraph discriminant analysis for hyperspectral image. IEEE Trans. Cybern. 49(7), 2404–2419 (2019)
16. Lu, H., et al.: Recovering low-rank tensor from limited coefficients in an ortho-normal basis using tensor-singular value decomposition. IET Signal Proc. 15(3), 162–181 (2020)
17. Miron, S., et al.: Tensor methods for multisensor signal processing. IET Signal Proc. 14(10), 693–709 (2020)
18. Lu, H., Plataniotis, K.N., Venetisapoulos, A.N.: MPCA: Multilinear principal component analysis of tensor objects. IEEE Trans. Neural Networks 19(1), 18–39 (2008)
19. Hu, R.X., et al.: Maximum margin criterion with tensor representation. Neurocomputing 73(12), 1541–1549 (2010)
20. Yan, S., et al.: Multilinear discriminant analysis for face recognition. IEEE Trans. Image Process. 16(1), 212–220 (2007)
21. 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)
22. Dai, G., Yeung, D.: Tensor embedding methods. In: Proceedings of National Conference on Artificial Intelligence, pp. 330–335 (2006)
23. Yan, S., et al.: Graph embedding and extensions: A general framework for dimensionality reduction. IEEE Trans. Pattern Anal. Mach. Intell. 29(1), 40–51 (2007)
24. Zhang, Z., Zhao, K.: Low-rank matrix approximation with manifold regularization. IEEE Trans. Pattern Anal. Mach. Intell. 35(7), 1717–1729 (2013)
25. Jiang, B., et al.: Image representation and learning with graph-Laplacian tucker tensor decomposition. IEEE Trans. Cybern. 49(4), 1417–1426 (2018)
26. Zhang, G.K., Ma, Z.M., Huang, H.D.: Dimensionality reduction of tensors based on local homeomorphism and global subspace projection distance minimum. IEEE Access 8, 116064–116077 (2020)
27. Zhou, G., et al.: Efficient nonnegative Tucker decompositions: Algorithms and uniqueness. IEEE Trans. Image Process. 24, 4990–5003 (2015)
28. Zhang, J., et al.: Low-rank regularized heterogeneous tensor decomposition for subspace clustering. IEEE Signal Process Lett. 25(3), 333–337 (2017)
29. Cai, D., et al.: Graph regularized nonnegative matrix factorization for data representation. IEEE Trans. Pattern Anal. Mach. Intell. 33(8), 1548–1560 (2010)
30. Das, S.: Hyperspectral image, video compression using sparse tucker tensor decomposition. IET Image Proc. 15(4), 964–973 (2020)
31. Gretton, A., et al.: Measuring statistical dependence with Hilbert-Schmidt norms. In: International Conference on Algorithmic Learning Theory, pp. 63–77. Springer, Berlin, Heidelberg (2005)
32. Kohler, M., Goldsby, S., Kasthurikumar, M.: Hilbert-Schmidt operator. In: Classes of Linear Operators, No. 1, pp. 138–147. Springer, Berlin, Germany (1990)
33. Gangeh, M.J., Zarkoob, H., Ghodsi, A.: Fast and scalable feature selection for gene expression data using Hilbert-Schmidt independence criterion. IEEE/ACM Trans. Comput. Biol. Bioinf. 14(1), 167–181 (2017)
34. Song, L., et al.: Supervised feature selection via dependency estimation. In: Proceedings of the 24th International Conference on Machine Learning, pp. 823–830 (2007)
35. Song, L., et al.: Feature selection via dependency maximization. J. Mach. Learn. Res. (13), 1393–1434 (2012)
36. Yan, K., Kou, L., Zhang, D.: Learning domain-invariant subspace using domain features and independence maximization. IEEE Trans. Cybern. 99, 1–12 (2017)
37. Dassad, B.B., Courty, N., Lefvre, S.: Sparse Hilbert Schmidt independence criterion and surrogate-kernel-based feature selection for hyper-spectral image classification. IEEE Trans. Geosci. Remote Sens. 55(4), 2385–2398 (2017)
38. Zhang, K., et al.: Covariate shift in Hilbert space: A solution via surrogate kernels. In: International Conference on Machine Learning, pp. 388–395 (2013)
39. Dodge, Y. (ed.) Statistical Data Analysis based on the L1-Norm and Related Methods. Birkhauser (2012)
40. Gangeh, M.J., Ghodsi, A., Kamil, M.S.: Kernelized supervised dictionary learning. IEEE Trans. Signal Process. 61(19), 4753–4767 (2013)
41. Gangeh, M.J., et al.: Multiview supervised dictionary learning in speech emotion recognition. IEEE/ACM Trans. Audio Speech Lang. Process. 22(6), 1056–1068 (2014)
42. Ma, Z., et al.: Nonlinear dimensionality reduction based on HSIC maximization. IEEE Access 6, 55537–55555 (2018)
43. Kreyszig, E.: Introductory Functional Analysis with Applications. Wiley, New York (1978)
44. Fu, Y., Huang, T.S.: Image classification using correlation tensor analysis. IEEE Trans. Image Process. 17(2), 226–234 (2008)
45. Cheng, L.X., et al.: Probabilistic tensor canonical polyadic decomposition with orthogonal factor. IEEE Trans. Signal Process. 65, 663–676 (2017)
46. Phan, A., et al.: On fast algorithms for orthogonal Tucker decomposition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6766–6770 (2014)
47. Phan, A., et al.: Damped Gauss-Newton algorithm for nonnegative Tucker decomposition. In: IEEE Statistical Signal Processing Workshop (SSP), pp. 665–668 (2011)
48. Kolda, T., Bader, B.: Tensor decompositions and applications. SIAM Rev. 51, 455–500 (2009)
49. Liu, X.H., et al.: Low-rank tensor completion for visual data recovery via the tensor train rank-1 decomposition. IET Image Proc. 14(11), 114–124 (2020)
50. Li, X., et al.: MR-NTD: Manifold regularization nonnegative Tucker decomposition for tensor data dimension reduction and representation. IEEE Trans. Neural Networks Learn. Syst. 28(8), 1787–1800 (2016)
51. Zhang, Z., Zha, H.: Principal manifold and nonlinear dimensionality reduction via tangent space alignment. SIAM J. Sci. Comput. 26(1), 313–338 (2004)
52. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computing 15(6), 1373–1396 (2003)
53. http://dataju.cn/Dataju/web/datasetInstanceDetail/62
54. http://people.csail.mit.edu/jjl/libpmk/samples/eth.html
55. https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
56. http://www.cs.toronto.edu/roweis/data.html
57. http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html
58. http://web.mit.edu/emeyers/www/facedatabases.html

How to cite this article: Niu, G., Ma, Z.: Tensor dimensionality reduction via mode product and HSIC. IET Image Process. 1–17 (2021).
https://doi.org/10.1049/ipr2.12285